Hardness of the maximum-independent-set problem on unit-disk graphs and prospects for quantum speedups

Ruben S. Andrist,[1] Martin J. A. Schuetz,[1,2] Pierre Minssen,[3] Romina Yalovetzky,[3] Shouvanik Chakrabarti,[3] Dylan Herman,[3] Niraj Kumar,[3] Grant Salton,[1,2,4] Ruslan Shaydulin,[3] Yue Sun,[3] Marco Pistoia,[3] and Helmut G. Katzgraber[1]

[1]*Amazon Quantum Solutions Lab, Seattle, Washington 98170, USA*
[2]*AWS Center for Quantum Computing, Pasadena, California 91125, USA*
[3]*Global Technology Applied Research, JPMorgan Chase, New York, New York 10017, USA*
[4]*California Institute of Technology, Pasadena, California 91125, USA*

(Received 31 July 2023; revised 22 August 2023; accepted 29 November 2023; published 21 December 2023)

Rydberg atom arrays are among the leading contenders for the demonstration of quantum speedups. Motivated by recent experiments with up to 289 qubits [Ebadi *et al.*, Science **376**, 1209 (2022)], we study the maximum-independent-set problem on unit-disk graphs with a broader range of classical solvers beyond the scope of the original paper. We carry out extensive numerical studies and assess problem hardness, using both exact and heuristic algorithms. We find that quasiplanar instances with Union-Jack-like connectivity can be solved to optimality for up to thousands of nodes within minutes, with both custom and generic commercial solvers on commodity hardware, without any instance-specific fine-tuning. We also perform a scaling analysis, showing that by relaxing the constraints on the classical simulated annealing algorithms considered in Ebadi *et al.*, our implementation is competitive with the quantum algorithms. Conversely, instances with larger connectivity or less structure are shown to display a time-to-solution potentially orders of magnitudes larger. Based on these results, we propose protocols to systematically tune problem hardness, motivating experiments with Rydberg atom arrays on instances orders of magnitude harder (for established classical solvers) than previously studied.

DOI: 10.1103/PhysRevResearch.5.043277

I. INTRODUCTION

Combinatorial optimization problems are pervasive across science and industry, with prominent applications in areas such as transportation and logistics, telecommunications, manufacturing, and finance. Given its potentially far-reaching impact, the demonstration of quantum speedups for practically relevant, computationally hard problems (such as combinatorial optimization problems) has emerged as one of the greatest milestones in quantum information science.

Over the past few years, programmable Rydberg atom arrays have emerged as a promising platform for the implementation of quantum information protocols [1–7] and (in particular) quantum optimization algorithms [8–14]. Some of the exquisite and experimentally demonstrated capabilities of these devices include the deterministic positioning of individual neutral atoms in highly scalable arrays with arbitrary arrangements [15,16], the coherent manipulation of the internal states of these atoms (including excitation into strongly excited Rydberg states) [17–19], the ability to coherently shuttle around individual atoms [20], and strong interactions mediated by the Rydberg blockade mechanism [12,21–23].

The physics of the Rydberg blockade mechanism has been shown to be intimately related to the canonical (NP-hard) maximum-independent-set (MIS) problem [8], in particular for unit-disk graphs. The MIS problem involves finding the largest independent set of vertices in a graph, i.e., the largest subset of vertices such that no edges connect any pair in the set; compare Fig. 1 for a schematic illustration. As shown in Ref. [8], MIS problems can be encoded with (effectively two-level) Rydberg atoms placed at the vertices of the target (problem) graph. Strong Rydberg interactions between atoms then prevent two neighboring atoms from being simultaneously in the excited Rydberg state, provided they are within the Rydberg blockade radius, thereby effectively implementing the independence constraint underlying the MIS problem. By virtue of this Rydberg blockade mechanism, Rydberg atom arrays allow for a hardware-efficient encoding of the MIS problem on unit-disk graphs, with the (tunable) disk radius $R_b \sim 1$–10 μm setting the relevant length-scale [6].

Overview of the main results

Recently, a potential (superlinear) quantum speedup over classical simulated annealing was reported for the MIS problem [12], based on variational quantum algorithms run on Rydberg atom arrays with up to 289 qubits arranged in two spatial dimensions. This work focused on benchmarking quantum variational algorithms against simulated annealing by viewing it as a classical analog of the adiabatic algorithm, yet left open the question of benchmarking against other state-of-the-art classical solvers. Motivated by this experiment, we perform a detailed analysis of the MIS problem on unit-disk graphs and assess problem hardness using both exact and heuristic methods. We provide a comprehensive algorithmic and numerical analysis, and we demonstrate the following: (i) Typical quasiplanar instances with Union-Jack-like connectivity (as studied in Ref. [12]) can be solved to optimality for up to thousands of nodes within minutes, with both custom and generic commercial solvers on commodity hardware, without any instance-specific fine-tuning. (ii) Systematic scaling results are provided for all solvers, displaying qualitatively better runtime scaling for solvers exploiting the quasiplanar problem structure than generic ones. In particular, we find that by relaxing the detailed balance constraint, and considering the low depth regime (both of which are required for analytic runtime lower bounds on SA described in Ref. [13]), our implementation of classical simulated annealing is competitive with the quantum algorithm's performance in Ref. [12]. (iii) Conversely, while the definition of problem hardness may be specific to the method used, instances with larger connectivity or less structure display a time-to-solution typically orders of magnitudes larger. (iv) Based on these results, we propose protocols to systematically tune problem hardness (as measured by classical time-to-solution), motivating experiments with Rydberg atom arrays on instances orders of magnitude harder (for established classical solvers) than previously studied.

This paper is organized as follows. In Sec. II we first formalize the problem we consider. Next, in Sec. III we describe the algorithmic tool suite with which we address this problem. In Sec. IV we then describe our numerical experiments in detail. Finally, in Sec. V, we draw conclusions and give an outlook on future directions of research.

II. PROBLEM SPECIFICATION

The MIS problem is a prominent combinatorial optimization problem with practical applications in network design [24], vehicle routing [25], and finance [26,27], among others, and it is closely related to the maximum clique, minimum vertex cover, and set packing problems [28].

A. Definition

Formally, the MIS problem reads as follows. Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, an independent set $S \subseteq \mathcal{V}$ is a subset of vertices of $\mathcal{G}$ such that no two vertices in S share an edge in $\mathcal{E}$. The maximum-independent-set problem is then the task of finding the *largest* independent set in $\mathcal{V}$. The cardinality of this largest independent set is referred to as the *independence number* |MIS|. One way to formulate the MIS problem mathematically is to first associate a binary variable $x_i \in \{0, 1\}$ to every vertex $i \in \mathcal{V}$, such that $x_i = 1$ if vertex $i = 1, \dots, N$ belongs to the independent set, and $x_i = 0$ otherwise. The MIS problem can then be expressed as a compact integer linear program of the form

$$\begin{aligned} \max \quad & \sum_i x_i \\ \text{s.t.} \quad & x_i + x_j \leqslant 1 \quad \forall (i, j) \in \mathcal{E}, \\ & x_i \in \{0, 1\}, \quad i = 1, \dots, N, \end{aligned} \tag{1}$$

with the objective to maximize the marked vertices while adhering to the independence constraint. Generalizations to the maximum-weight independent set problem are straightforward [25].

A formulation of the MIS problem that is commonly used in the physics literature expresses the integer linear program in Eq. (1) in terms of a Hamiltonian that includes a (soft) penalty to nonindependent configurations (i.e., when two vertices in the set are connected by an edge) [8]. This Hamiltonian is given by

$$H = -\sum_i x_i + V \sum_{(i,j)\in\mathcal{E}} x_i x_j, \tag{2}$$

with a negative sign in front of the first term because the largest independent set is searched for within a minimization problem, and where the penalty parameter V enforces the constraints. Energetically, this Hamiltonian favors having each variable in the state $x_i = 1$ unless a pair of vertices are connected by an edge. For $V > 1$, the ground state is guaranteed to be a MIS, because it is strictly more favorable to have at most one vertex per edge in the set as opposed to both vertices being marked [12]. Still, within this framework, the independence constraint typically needs to be enforced via postprocessing routines (as is done, for example, in Ref. [12]). Mapping the binary variables x_i to two-level Rydberg atoms subject to a coherent drive with Rabi frequency Ω and detuning Δ, one can then search for the ground state of the Hamiltonian H (encoding the MIS) via, for example, quantum-annealing-type approaches using quantum tunneling between different spin configurations [8,12].

B. Problem hardness

The MIS problem is known to be strongly NP-hard, making the existence of an efficient algorithm for finding the maximum independent set on generic graphs unlikely. As such, the MIS problem is even hard to approximate [29], and in general it cannot be approximated to a constant factor in polynomial time (unless $P = NP$).

Here, however, we focus primarily on the MIS problem on unit-disk graphs (dubbed MIS-UD hereafter), given their intimate relation to Rydberg physics [8,12]. Our main goal is to empirically assess the hardness of MIS-UD. As schematically depicted in Fig. 1, unit-disk graphs are defined by vertices on a two-dimensional plane with edges connecting all pairs of vertices within a unit distance. The MIS-UD problem appears in practical situations with geometric constraints such as map labeling [30] and wireless network design [24]. While approximate solutions to MIS-UD can be found in polynomial time [31], solving the problem *exactly* is still known to be NP-hard for worst-case instances [12,32].

C. Problem instances and figures of merit

The unit disk (UD) problem instances of interest can be characterized by the number of nodes N, the side length of the underlying square lattice L, and the filling fraction ϱ, with $N \approx \varrho L^2$, as schematically depicted in Fig. 1. Following Ref. [12], we focus on single-component (nonplanar, but quasiplanar) UD instances with nearest and next-nearest

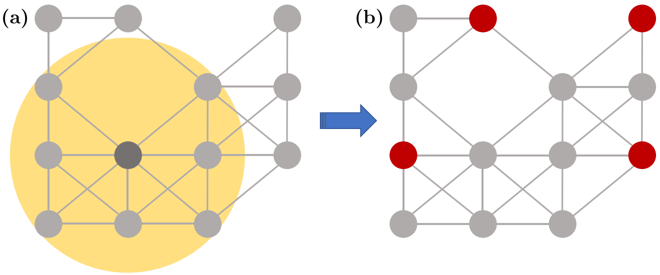

FIG. 1. Schematic illustration of the problem. (a) We consider unit-disk graphs with nodes arranged on a two-dimensional square lattice with lattice spacing a and filling fraction $\varrho \sim 80\%$, and edges connecting all pairs of nodes within a unit distance (illustrated by the circle). For $\sqrt{2}a \leqslant R_b < 2a$ (as considered here), nodes are connected to nearest and next-nearest neighbors resulting in a (quasi-planar) Union-Jack pattern with maximum degree $d_{\max} = 8$. (b) Our goal is to solve the MIS problem on this family of instances (as depicted here with nodes colored in red in the right panel) and assess the hardness thereof using both exact and heuristic algorithms.

(diagonal) couplings only, resulting in Union-Jack-type graphs with maximum degree $d_{\max} = 8$. Accordingly, these instances consist of (at maximum) $n_{\text{corner}} \leqslant 4$ *corner*, $n_{\text{boundary}} \leqslant 4(L-2)$ *boundary*, and $n_{\text{bulk}} \leqslant (L-2)^2$ *bulk* nodes, with (at maximum) three, five, and eight neighbors, respectively, and a total of at most $|\mathcal{E}|_{\max}$ edges, with $|\mathcal{E}|_{\max} = 4L^2 - 6L + 2$. Because $|\mathcal{E}|_{\max} \sim N$, the graph density D_{graph} scales as $D_{\text{graph}} = 2|\mathcal{E}|_{\max}/N(N-1) \sim 1/N$, showing that these instances become sparser as the system size N grows. If not otherwise specified, we take $\varrho = 80\%$, as was done in Ref. [12]. For comparison, we also run experiments on (unstructured) random Erdős-Rényi (ER) graphs denoted as $\mathcal{G}(n, m)$, chosen uniformly at random from the collection of all graphs with n nodes and m edges, or similarly $\mathcal{G}(n, p)$ for graphs constructed by connecting nodes randomly with probability p.

To assess and compare the performance of various algorithms (as specified below), we consider the following figures of merit. We use |MIS| to denote the independence number, while P_{MIS} refers to the probability of observing an (exact) MIS within a fixed number of steps [12]. For a given instance, many MIS solutions may be available, with the corresponding number of MIS solutions (i.e., the MIS degeneracy) denoted as D_{MIS}. Similarly, the quantity $D_{\text{MIS}-1}$ refers to the number of first excited states (i.e., independent sets of size $|\text{MIS}| - 1$). As shown in Ref. [12], in the context of simulated annealing, problem hardness may further be specified in terms of the conductance-like hardness parameter

$$\mathbb{H} = \frac{D_{\text{MIS}-1}}{|\text{MIS}| \cdot D_{\text{MIS}}}, \tag{3}$$

with the factor $|\text{MIS}| D_{\text{MIS}}$ denoting the number of possible transitions from a first excited state into a MIS ground state. Finally, we are interested in the *time-to-solution* (TTS). For exact methods, TTS refers to the time needed to find the optimal solution (i.e., ground state). While the optimum may be found after time TTS, additional time may be required to provide an optimality certificate, resulting in the *time-to-optimality* (TTO) timescale, with TTO $\geqslant$ TTS. By definition, provable optimality is not available with heuristic methods. Here, we define TTS_{99} as the time required to find the exact solution (ground state) with 99% success probability. We can then write TTS_{99} as

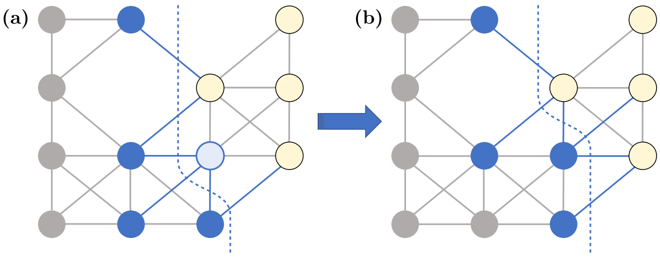

FIG. 2. Schematic illustration of the (exact) sweeping line algorithm (SLA) as applied to the MIS-UD problem. (a) SLA proceeds by sweeping a fictitious line across the graph and tracking all potentially optimal MIS configurations on this boundary, efficiently exploiting the quasiplanar structure of the UD graph. Processed nodes are shown in gray, boundary nodes in blue, and unprocessed nodes in yellow. (b) The light blue node from (a) is added to the boundary, while the bottom left blue node is dropped (as a result of not having any more connections to unprocessed nodes).

$$\text{TTS}_{99} = \tau R_{99}, \tag{4}$$

where τ refers to the time of a single run (shot), and

$$R_{99} = \frac{\log(1-0.99)}{\log(1-P_{\text{MIS}})} \tag{5}$$

is the number of shots (repetitions) needed to reach the desired success probability [33]. For small values of P_{MIS}, we have $R_{99} \approx 4.6/P_{\text{MIS}}$, showing that the success probability of a *single* run P_{MIS} determines the inverse of the time-to-solution for heuristic algorithms.

III. ALGORITHMIC TOOL SUITE

In this section, we detail the algorithms used to solve the MIS problem on UD graphs. We distinguish between *exact* methods (which by design can deterministically find the ground state, typically at the expense of an exponential runtime) and *heuristics* (which cannot provide an optimality certificate, but may require shorter runtimes).

A. Exact methods

1. Sweeping line algorithm

We first consider an exact sweeping line algorithm (SLA) [34] that efficiently exploits the quasiplanar structure of the UD instances considered here. The anatomy of the SLA is schematically illustrated in Fig. 2. The SLA is based on full enumeration and works by sweeping a fictitious line across the two-dimensional plane (e.g., from left to right). Specifically, the algorithm proceeds as follows. We define the *boundary* as the set of all processed nodes that still share an edge with an unprocessed node. At each step, we track the size of the largest independent set (so far) for each valid boundary variant (set of assigned nodes on the boundary). As the line is swept across the graph, it stops at every node $i = 0, \dots, N-1$. We then generate the new variants at step i from those at step $i-1$ as follows:

(i) If the variant has a neighboring node of i assigned, we generate only a new variant with i not selected (that is, $x_i = 0$).

(ii) Otherwise we also create a new variant with i assigned as $x_i = 1$ (which increases its independent set).

Note that forward-looking information is not required, and only boundary nodes are relevant for this decision. Once the new variants have been generated, node i becomes part of the boundary, and we proceed with the next step. When moved across the graph, this recipe generates all valid sets with runtime $O(2^N)$. However, we can efficiently summarize information that is not relevant to finding the MIS:

(i) Adding new nodes typically removes older nodes from the boundary (because they no longer have a connection to an unprocessed node); cf. Fig. 2.

(ii) We only need to track the size of the largest independent set for each boundary variant (i.e., for any boundary configuration we can discard any option with an equal or smaller number of assignments).

For the UD graphs considered here, the number of variants tracked on any given boundary is limited by the number of valid assignments on the boundary, #MIS. When processing the nodes in order, the boundaries form continuous (mostly) one-dimensional strips and $\#\text{MIS}(L) \leqslant \text{Fib}(L+1)$ (which can be shown by induction). As a result, the memory requirement for SLA scales as $O_{\text{mem}}(\text{Fib}(L)) = O_{\text{mem}}(\text{Fib}(\sqrt{N}))$ (to hold the variants at each step). SLA finds the optimal solution after N steps (TTS = TTO), processing all variants at each step with a runtime of $O(N\text{Fib}(L)) = O(N\text{Fib}(\sqrt{N})) \approx O(N\varphi^{\sqrt{N}})$, where $\varphi \approx 1.62$ is the golden ratio. This procedure can also be modified to count the degeneracy of the ground and first excited state (i.e., D_{MIS} and $D_{\text{MIS}-1}$, respectively) by adjusting summarization accordingly.

2. *Branch & bound solvers*

We complement our custom SLA solver with commercial solvers based on the branch and bound (B&B) search method. In particular, we use the solver offered by CPLEX [35]; similar results were observed with Gurobi [36]. In practice, these solvers are among the *de facto* go-to tools for many hard, mixed-integer optimization problems. By design, B&B solvers provide upper and lower bounds on the solution, with the difference between these yielding an *optimality gap*, thereby giving information about the quality of the solution (at any step throughout the algorithmic evolution). Assuming a maximization problem, the lower bound corresponds to the best known feasible solution, whereas the upper bound refers to the optimal value for the corresponding relaxed problem in the B&B procedure. In this work, we focus on the TTS and TTO figures of merit, which are readily provided by our chosen solvers. We evaluate TTO by enforcing a zero gap between the upper and the lower bound, and TTS by setting the upper bound to be the optimal solution found previously. Thus, the solver terminates successfully as soon as it reaches the optimal solution. Typically, we find TTO is strictly greater than TTS because of additional time required to prove optimality. To draw a clear line between the B&B solvers and the heuristic solvers described below, we deactivate the B&B solvers' capability to find feasible solutions heuristically. As such, in practice we expect smaller values for TTS when utilizing these additional features of modern B&B solvers, effectively making the TTS values we report here upper *bounds* for B&B based performance. Finally, to account for the multithreading capabilities of B&B solvers, we report the *process time*, i.e., the sum of system and user CPU seconds of each core used during the calculation, rather than the *wall-clock time*. We checked that the multithreading overhead does not impact the O complexities inferred from the numerical experiments; see Appendix A 2 for more details.

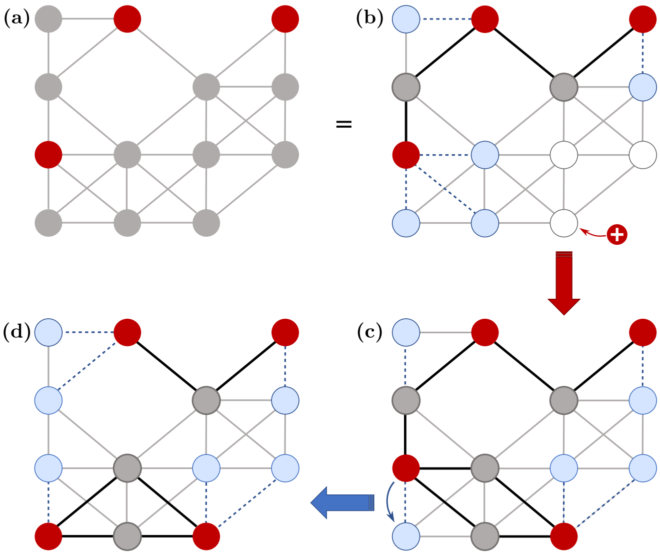

FIG. 3. Schematic illustration of the heuristic simulated annealing (SA) solver. The original configuration (a) is overlaid with connectivity statistics in (b): nodes in the set (red), nodes without marked neighbors (white), and nodes with a single neighbor (blue). Potential moves are (i) removal of (red) nodes currently in the set, (ii) addition of currently white nodes to the set, and (iii) swapping a blue node with its red neighbor. Gray nodes have more than one adjacent node in the set and no valid moves. From (b) to (c), one node is added to the set and the statistics are updated accordingly. From (c) to (d), one blue node is swapped with its adjacent red node.

B. Heuristics

Apart from the exact methods outlined above, we utilize two established (physics-inspired) heuristic algorithms [37], namely simulated annealing (SA) and parallel tempering (PT). For a schematic illustration, see Fig. 3. In these Markov chain Monte Carlo (MCMC) samplers, a random modification to the current solution is proposed at each step of the algorithm and accepted depending on its effect on a specified figure of merit. For the MIS-UD problem at hand, we use the size of the independent set (IS) as the figure of merit to optimize. Any proposed move (update) is then accepted with a probability governed by a temperature parameter $T > 0$ according to the Metropolis acceptance criterion [38]:

$$p_{\text{accept}} = \begin{cases} 1 & \text{if } \Delta_{\text{IS}} \geqslant 0, \\ e^{\Delta_{\text{IS}}/T} & \text{otherwise.} \end{cases}$$

That is, moves that *increase* the size of the IS (i.e., $\Delta_{\mathrm{IS}} > 0$) are always accepted, while those reducing its size are suppressed—at first only marginally at high temperatures (during initial exploration), but then heavily at low temperatures T (during final exploitation). Our Markov dynamics consist of individual additions/removals and swaps of neighboring sites. We ensure the independence criterion is never violated by continuously tracking the full list of valid moves. The random selection from this set is then biased towards adding nodes and performing swaps in order to increase the acceptance rate (while accounting for the shift in energy scales by adapting the cooling schedule).

Simulated annealing (SA) aims to find a high-quality solution by starting from a random initial solution and initially high temperature to then gradually lower $T \rightarrow 0$ (according to some annealing schedule) until no further improvement is seen [39]. This allows the system to first explore the solution space while the temperature is high, but eventually drives the state into a nearby (local, potentially global) optimum. The cooling schedule is optimized to quickly identify this local optimum, and frequent restarts (as specified by the parameter `num_restarts`) from different initial positions are used to increase the chance of finding the global optimum. We note that our implementation of SA differs from the one presented in Ref. [12] in several ways: (i) Our implementation is not based on a (soft) penalty model as described by Eq. (2), but rather involves only moves compatible with the (hard) independence criterion (such that only the feasible space is searched). (ii) Proposal probabilities are biased towards additions and exchanges to increase acceptance. (iii) We use a geometric cooling schedule in combination with frequent restarts, as opposed to a constant low temperature as used in Ref. [12]. In particular, we note that our implementation of SA breaks detailed balance, for the sake of improved performance.

Parallel tempering (PT) attempts to efficiently explore the solutions space by simulating several Markov chains concurrently at different temperatures [40–42]. Exchange moves allow for swapping of states between neighboring chains (in temperature space) such that the best solutions are shuffled to lower temperatures for further local optimization. At the same time, less promising candidates are moved to higher temperatures, where large-scale restructuring is possible.

In the following, we will focus on SA, since our implementation of PT did not provide any substantial performance benefits over SA for the problem instances studied here. This is likely because of the relatively fast identification of local minima by SA (within tens to hundreds of sweeps) compared to the mixing time needed to exploit the benefits of PT; as such it is more efficient to restart at a random position than to invest in overcoming local energy barriers.

IV. NUMERICAL EXPERIMENTS

We now turn to our numerical results. We report on the TTS for all exact and heuristic algorithms described above, as a function of system size N and hardness parameter $\mathbb{H}$, with the goal to provide a comprehensive assessment of the hardness of random MIS-UD problem instances. For reference, we also study the MIS problem on similar yet less structured instances (with the same number of nodes and edges), and we provide protocols to systematically tune problem hardness (as measured by TTS) over several orders of magnitude. The classical hardware on which our numerical experiments were run is specified in Appendix A.

A. Scaling with the problem size

We first report on TTS for the MIS-UD problem as a function of system size, given by the number of nodes $N \approx \varrho L^2$ at fixed density $\varrho = 0.8$. Note that we have run a few additional experiments for different values of ϱ, with $\varrho \approx 0.8$ providing one of the hardest problems for Union-Jack connectivity (as evidenced by the largest TTS). Thus, we primarily focus on $\varrho = 0.8$, following Ref. [12]. As shown in Fig. 4, we find that the MIS-UD problem can be solved to optimality (with both the custom SLA and the generic B&B solvers) for hundreds of nodes in subsecond timescales. Larger instances with up to thousands of nodes can still be solved to optimality within minutes on commodity hardware, without any instance-specific fine-tuning. For the exact SLA solver we infer a runtime scaling of $\mathrm{TTS}_{\mathrm{SLA}} = O(N\phi^{\sqrt{N}})$, where $\phi \approx 1.62$ is the golden ratio (expected for the scaling of the Fibonacci sequence) and the notable $\sqrt{N}$ dependence in the exponent is attained at the expense of exponential memory requirements (as discussed above). For the B&B solver, we obtain $\mathrm{TTS}_{\mathrm{B\&B}} = O(2^{0.0045N})$. Similarly, as shown in Fig. 5, we find that UD instances with hundreds of nodes (i.e., $L \leqslant 25$) can typically be solved efficiently in subsecond timescales with the SA heuristic. For the 2% most difficult instances, we observe a scaling of $\mathrm{TTS}_{99} = O(2^{0.0128N})$ for SA. However we also observe a relatively large spread spanning several orders of magnitude in TTS_{99} (in particular when compared to the results obtained with SLA), thus motivating a more detailed analysis of problem hardness, as discussed next.

B. Scaling with the hardness parameter

Some of the results presented in the previous section display significant instance-to-instance variations in TTS_{99}, potentially spanning several orders of magnitude, even for a fixed system size N. As argued in Ref. [12], these variations may largely be due to (large) differences in the total number of MIS solutions, given by the ground-state degeneracy D_{MIS}, a quantity that can be calculated either with tensor-network methods [43] or within the exact SLA method outlined above (at least for small to intermediate system sizes up to $N \sim 1000$). Intuitively, the less degenerate the ground state, the harder it is to hit the global optimum. In particular, based on experiments with up to $N = 80$ qubits in Ref. [12], a quantum speedup over classical SA has been reported in the dependence of the success probability P_{MIS} on the hardness parameter $\mathbb{H}$ that accounts for both the degeneracy of the ground as well as first excited states (denoted by D_{MIS} and $D_{\mathrm{MIS}-1}$, respectively), as given in Eq. (3).

We now follow Ref. [12] and consider algorithmic performance in terms of this hardness parameter $\mathbb{H}$ for values as high as $\mathbb{H} \sim 10^8$, complementing existing results based on classical SA [12] with results for the SLA, and B&B solvers. Note that hyperparameter optimization has been performed

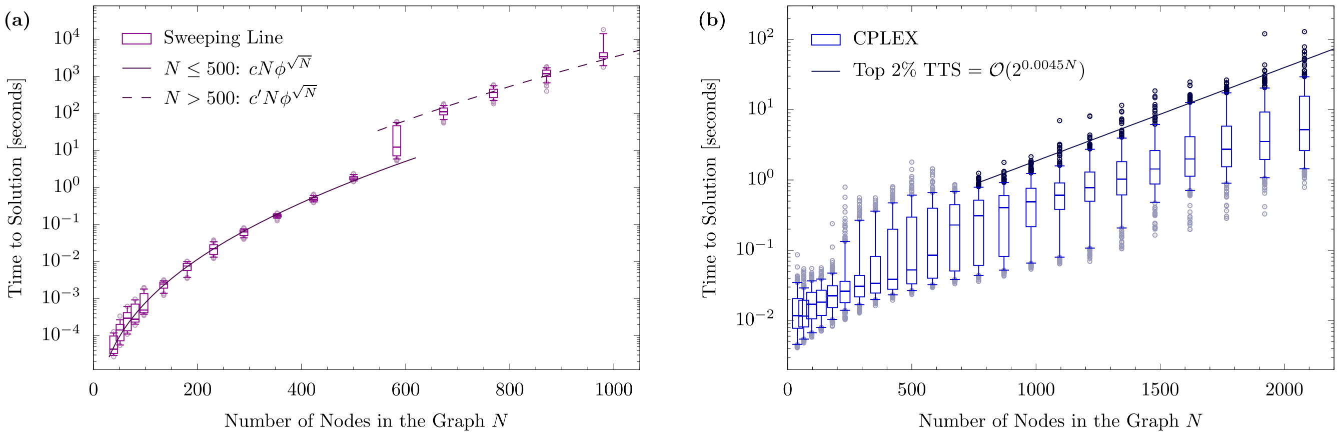

FIG. 4. Time-to-solution (TTS) for the exact solvers. (a) TTS for the exact SLA solver as a function of system size N. For every system size N, 1000 random UD instances with $\varrho = 0.8$ have been considered. The data fit reasonably to $\mathrm{TTS}(N) \approx cN\phi^{\sqrt{N}}$, where the basis of $\phi \approx 1.62$ is the theoretical expectation for the Fibonacci sequence. At larger system sizes ($N > 500$), high memory usage causes slower access times (cache misses), resulting in a substantially larger prefactor c'. (b) TTS for the B&B solver as a function of system size N. For every system size N, 1000 random UD instances have been considered; see Appendix A 3 for a box plot description. Problems with hundreds (thousands) of nodes can be solved to optimality in subsecond (minute) timescales. The solid line is the linear regression over instances whose TTS are in the top highest 2%. The linear regression minimizes the residual sum of squares of log(TTS).

for our heuristic solvers, although without any instance-to-instance fine-tuning. For direct comparison, the results for the exact SLA as well as the heuristic SA solvers are displayed in Fig. 6 showing a remarkably different behavior. Qualitatively, we find that TTS_{99} for the SA solver displays a strong dependence on the hardness parameter $\mathbb{H}$, in line with results reported in Ref. [12]. Conversely, virtually no dependence between TTS and $\mathbb{H}$ is observed for the exact SLA solver, as expected, thereby demonstrating that the conductance-like hardness parameter $\mathbb{H}$ successfully captures hardness for algorithms undergoing Markov-chain dynamics. Alternative algorithmic paradigms such as sweeping line or branch and bound, however, may require a different notion of hardness. Similarly, for the B&B solvers we find that TTS is (weakly) correlated with the hardness parameter $\mathbb{H}$, although mostly because of their common correlation with the system size N. Specifically, using linear regression, we have found that the partial correlation of $\log_{10}(\mathbb{H})$ and $\log_{10}(\mathrm{TTS})$ (controlling for system size) is smaller than 0.05; see Appendix B 2 for further details. This weak correlation suggests that (similarly to our SLA results) the hardness parameter $\mathbb{H}$ does not appear to be a reliable measure of hardness for B&B-type solvers.

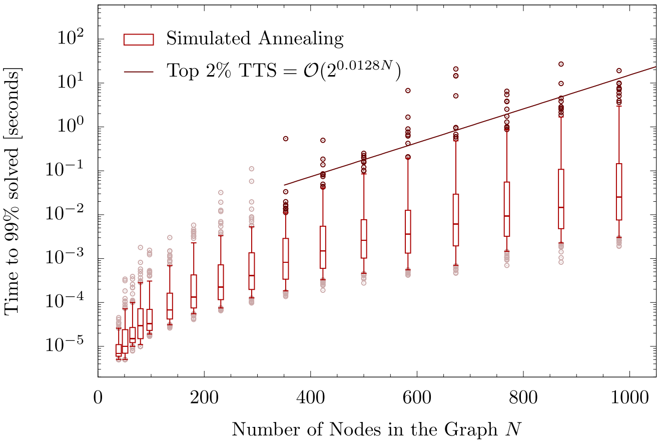

FIG. 5. Time required to reach 99% success probability (TTS_{99}) for the heuristic SA solver as a function of system size N (i.e., how long the solver should run for a 99% chance of finding the optimal solution). For every system size N, 1000 random UD instances at $\varrho = 0.8$ filling have been considered; see Appendix A 3 for a box plot description. The solid line is the linear regression over instances whose TTS are in the top highest 2%. The linear regression minimizes the residual sum of squares of log(TTS).

Finally, we complement the TTS results above with results for the success probability P_{MIS} as a function of the hardness parameter $\mathbb{H}$ for the SA solver, as done in Ref. [12] for both SA and quantum algorithms for instances with hardness of up to $\mathbb{H} \sim 10^{3}$. Our results with hardness of up to $\mathbb{H} \sim 10^{8}$ are shown in Fig. 7. Following Ref. [12], for fixed depth (i.e., number of SA sweeps), fits are provided assuming the functional form $P_{\mathrm{MIS}} = 1 - \exp(-C\mathbb{H}^{-\alpha})$, where C refers to a positive fitted constant that (in general) could have a polynomial dependence on the system size N, and smaller values of α yield a larger success rate P_{MIS}. While there are significant variations in the data, on average we observe a scaling $P_{\mathrm{MIS}} \approx 1 - \exp(-C\mathbb{H}^{-0.66})$ (solid black line), i.e., $\alpha = 0.66$; additional results with size-dependent depth (not shown) suggest an even higher success probability P_{MIS} but preclude simple fits because of additional size-dependent effects in the data. In particular, a fixed depth of 32 is arguably too small for the largest instances considered here, but was chosen nevertheless for better comparison with results reported in Ref. [12]. For comparison, a fit with the exponent $\alpha = 1.03$ was reported for SA in Ref. [12]. If one restricts the analysis to graphs with minimum energy gaps sufficiently large to be resolved in the duration of the (noisy) quantum evolution, the optimized quantum algorithm demonstrated in Ref. [12] was shown to fit best to $\alpha = 0.63$, i.e., comparable to $\alpha = 0.66$ as found with our implementation of classical SA.

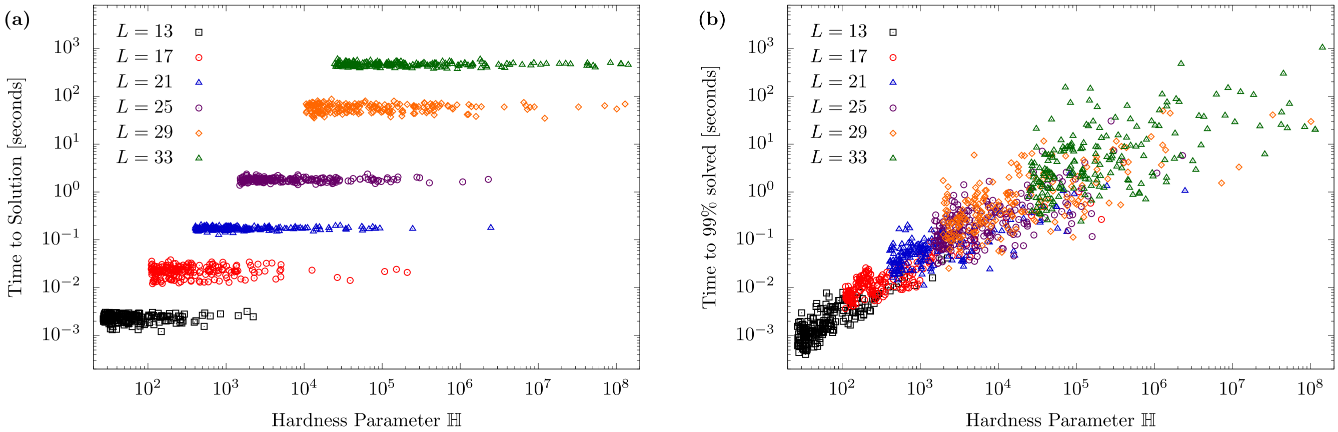

FIG. 6. (a) Time-to-solution (TTS) for the exact SLA solver as a function of the hardness parameter $\mathbb{H}$. Virtually no dependence on $\mathbb{H}$ is observed, showing that TTS is fully determined by the system size $N \sim L^2$. (b) Conversely, for the Markov-chain based SA solver, TTS$_{99}$ shows a strong correlation with the hardness parameter $\mathbb{H}$, as expected.

C. Beyond Union-Jack connectivity

To provide further context for the results reported above, we now study hardness as we gradually change the topology of the problem instances. Specifically, going beyond Union-Jack-type instances (with $\sqrt{2} \leqslant r < 2$ fixed) as studied so far, we analyze TTS following two protocols by either (i) systematically tuning the blockade radius $r = R_b/a$, or (ii) randomly rewiring edges of the graph. While protocol (i) prepares UD graphs (with varying connectivity), protocol (ii) explicitly breaks the UD structure via random (potentially long-range) interactions, ultimately preparing random (structure-less) ER graphs. The results of this analysis are shown in Figs. 8 and 9, respectively. We find that problem hardness (as measured here by TTS for the established B&B solver) can be tuned systematically over several orders of magnitude.

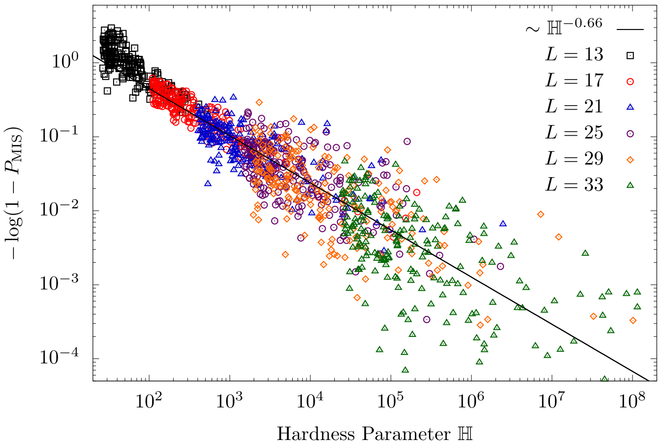

FIG. 7. Estimated success probability P_{MIS} for the heuristic SA solver as a function of the hardness parameter $\mathbb{H}$. Here we plot $-\log(1 - P_{\mathrm{MIS}})$ for SA with a fixed depth of 32, for UD graphs selected from the top two percentile of hardness parameter $\mathbb{H}$ for each system size $L = 13, \ldots, 33$. Hollow points represent our SA implementation with bias moves in valid configuration space. Power-law fits to the form $\sim\mathbb{H}^{-\alpha}$ are used to extract scaling performance with graph hardness $\mathbb{H}$.

As shown in Fig. 8, we find that the MIS-UD problem is relatively easy for both small ($r < 2$) and large radii (as expected because the MIS problem is trivial for both edgeless and complete graphs), but significantly harder in between, with a pronounced peak at $r = 3$. For example, for $L = 21$ and $\varrho = 0.8$ (giving $N \approx 350$) we observe an increase in TTS over two to three orders of magnitude for $r = 3$ instances compared to instances with small or large radii. This behavior appears to be generic, as we have observed similar behavior with the SLA solver (cf. Fig. 12), again showing pronounced peaks in TTS for $r = 2, 3, 4$; see Appendix B for more details. This observation may be attributed to a density-of-states-like effect: While the boundary size grows monotonically with r, we find that the number of boundary variants peaks at $r = 2, 3, 4$ before decreasing with r as the (feasible) state space becomes smaller, thus correlating the observed TTS behavior with the number of boundary variants. This spread in TTS is found to increase with system size, as shown in Fig. 8(b) for instances with Union-Jack topology (with $r = \sqrt{2}$) as well as instances with $r = 3$.

Alternatively, following protocol (ii) the MIS problem appears to become orders of magnitude harder when randomly rewiring edges (thereby breaking the UD structure), as shown in Fig. 9. In particular, we find that structureless ER graphs can yield a TTS orders of magnitudes larger than more structured UD graphs (with the same average number of nodes and edges), in agreement with similar results for random UD graphs where vertices are placed randomly (and not on a square lattice) in a two-dimensional box with some fixed density [8].

D. Implementation with Rydberg arrays

The two protocols outlined above may be implemented in future experiments with Rydberg atom arrays, either by (i) tuning the Rydberg blockade radius R_b and/or the lattice spacing a, or by (ii) implementing embedding schemes with ancilla Rydberg chains, as proposed in Refs. [44,45], thus suggesting a potential recipe to benchmark quantum algorithms on instances orders of magnitude harder (for established classical solvers) than previously studied. For example, for

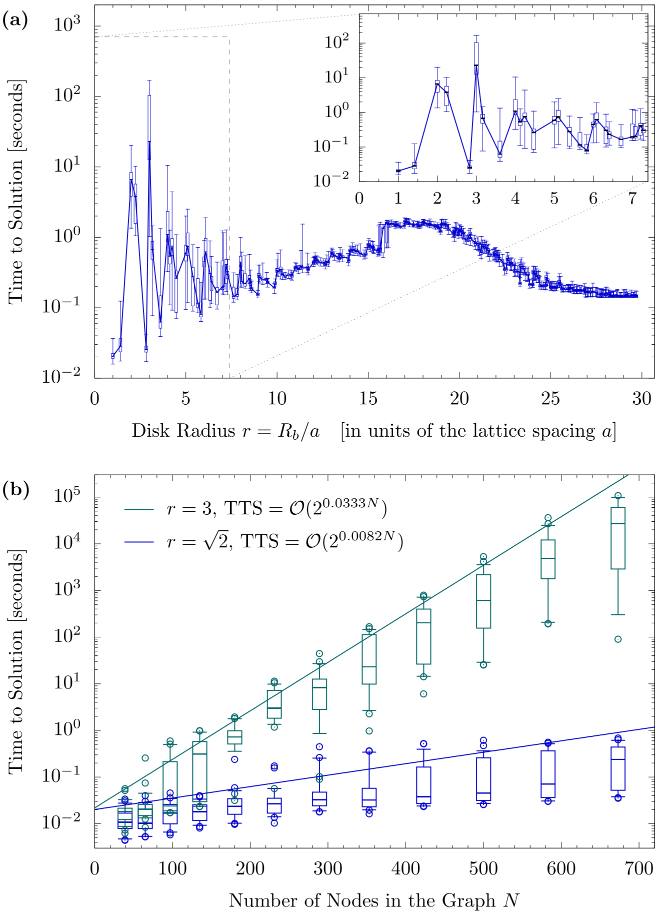

FIG. 8. (a) Hardness transition as a function of the disk radius (in units of the lattice spacing) $r = R_b/a$, as given by the time-to-solution (TTS) for the B&B solver, shown here for system size $L = 21$ and density $\varrho = 0.8$ (i.e., $N \approx 350$), with 100 random seeds per radius. (b) TTS as a function of system size $N = \varrho L^2$ for $r = \sqrt{2}$ (blue) and $r = 3$ (green), the latter referring to the pronounced peak observed in (a). The solid lines are a linear regression fit over 100 instances with TTS in the highest 10%, with corresponding R^2 values of 0.87 and 0.98 for $r = \sqrt{2}$ and $r = 3$, respectively. Instances with $r = 3$ appear to be much harder than those with $r = \sqrt{2}$. See Appendix A 3 for a box plot description.

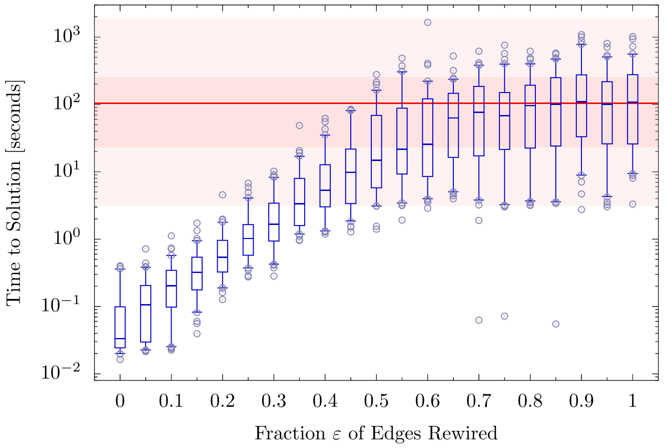

FIG. 9. Hardness transition from unit-disk (UD) to random Erdős-Rényi (ER) graphs. Time-to-solution (TTS) for MIS as a function of the fraction ε of edges rewired, with 150 random seeds. Starting from Union-Jack-type UD graphs (left), edges are randomly selected and rewired, thereby gradually breaking the UD connectivity, and ultimately generating random ER graphs (right). While the original UD graphs can be solved to optimality in $\sim 10^{-2}$ s, comparable ER graphs (with the same number of nodes and edges) display a TTS orders of magnitude larger. The red line and the two shaded areas refer to the median TTS over 500 instances for MIS on random ER graphs, the TTS among the 25% and 75% and the minimum and maximum whiskers, respectively. Numerical parameters: $L = 21$ and $\varrho = 80\%$.

today's Rydberg atom arrays (such as the QuEra Aquila device available through Amazon Braket), we estimate that $R_b/a \sim 3$ (amounting to a maximum degree of $d_{\max} \sim 28$) should be achievable already today, with potentially even larger values enabled by future hardware improvements. Experiments like these may also provide new insights into effects stemming from the long-range interaction tails associated with the Rydberg interactions.

E. Prospects for quantum speedups

With the goal to help identify regimes and system sizes where quantum algorithms could be useful, we now briefly revisit our results in light of ongoing efforts towards quantum advantage. Adopting the taxonomy put forward in Ref. [46], within a larger hierarchy of potential quantum speedups, the quantum speedup demonstrated in Ref. [12] could be classified as *limited sequential quantum speedup*, as it was obtained by comparing a quantum annealing type algorithm over a particular implementation of the classical (sequential) simulated annealing algorithm (that was designed to fulfill detailed balance). Here, we have tried to extend the classical SA benchmarking results (by pushing the hardness parameter up to $\mathbb{H} \sim 10^8$), with a different implementation of SA that breaks detailed balance but shows better performance. While our SA-based scaling results hint at performance similar to the quantum algorithm's performance in Ref. [12], we note that the corresponding exponent shows a dependence on the somewhat arbitrary cutoff in hardness and additional dependence on system size N if the depth is not fixed; the details thereof will be analyzed in future research. Still, within the aforementioned hierarchy of quantum speedups, our analysis points to a potential next milestone, in the form of the experimental demonstration of a (more general) *limited non-tailored quantum speedup*, by (for example) comparing the performance of the quantum algorithm to the best-known generic classical optimization algorithm. In particular, B&B solvers (as studied here) could be good candidates to fill the role for the latter, and the protocols outlined above motivate potential quantum experiments for such studies. Again, for reference for the MIS-UD problem on Union-Jack instances, empirically for most instances (taken as 98th percentile) we can upper-bound the TTS needed by classical B&B solvers through a runtime scaling of TTS $= O(1.0031^N)$ (cf. Fig. 4), setting an interesting, putative bar for quantum algorithms to beat.

V. CONCLUSION AND OUTLOOK

In summary, we have studied the maximum-independent-set problem on unit-disk graphs (as it can be encoded efficiently with Rydberg atom arrays [8,12]) using a plethora of exact and heuristic classical algorithms. We have assessed problem hardness, showing that instances with thousands of nodes can be solved to optimality within minutes using both custom and generic commercial solvers on commodity hardware, without any instance-specific fine-tuning. We have also performed a detailed scaling analysis, showing that our implementation of classical simulated annealing is competitive with the quantum algorithm's performance in Ref. [12]. Finally, we have devised protocols to systematically tune problem hardness over several orders of magnitude. In future studies, the problem hardness may be tuned even further by generalizing our work from unweighted to weighted graphs, thereby potentially lifting the ground-state degeneracy, as could be studied with local detuning control in Rydberg atom arrays. We hope that these protocols may trigger interesting future experiments further exploring the hardness of the MIS problem with Rydberg atom arrays as pioneered in Ref. [12].

An open source demo version of our code is publicly available in Ref. [47].

ACKNOWLEDGMENTS

We thank Maddie Cain, Alex Keesling, Eric Kessler, Harry Levine, Mikhail Lukin, Hannes Pichler, and Shengtao Wang for fruitful discussions. We thank Peter Komar and Gili Rosenberg for detailed reviews of the manuscript, and Marjan Bagheri, Victor Bocking, Alexander Buts, Lou Romano, Peter Sceusa, and Tyler Takeshita for their support.

R.S.A., M.J.A.S., P.M., and R.Y. contributed equally to this work. M.P. and H.G.K. acted as co-equal principal investigators (co-PIs).

This paper was prepared for informational purposes with contributions from the Global Technology Applied Research center of JPMorgan Chase & Co. This paper is not a product of the Research Department of JPMorgan Chase & Co. or its affiliates. Neither JPMorgan Chase & Co. nor any of its affiliates makes any explicit or implied representation or warranty and none of them accept any liability in connection with this paper, including, without limitation, with respect to the completeness, accuracy, or reliability of the information contained herein and the potential legal, compliance, tax, or accounting effects thereof. This document is not intended as investment research or investment advice, or as a recommendation, offer, or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction.

APPENDIX A: ADDITIONAL INFORMATION

In this Appendix, we provide further details for selected aspects discussed in the main text. In Appendix A 1 we derive the maximum and expected number of edges for Union-Jack-like graphs with density ϱ. In Appendix A 2 we provide details on the hardware on which our numerical experiments were run. In Appendix A 3 we provide additional information on our plots. In Appendix A 4 we derive the scaling of TTS with the hardness parameter $\mathbb{H}$ for SA.

1. Expected number of edges in UD graphs

In this section, we show that for quasi-planar UD graphs with Union-Jack-type connectivity and filling fraction ϱ (as considered here), the expected (or average) number of edges amounts to $\overline{|\mathcal{E}|} = \varrho^2(4L^2 - 6L + 2)$; for full filling of the square lattice at $\varrho = 100\%$ we recover $\overline{|\mathcal{E}|} \rightarrow |\mathcal{E}|_{\max} = 4L^2 - 6L + 2$. To this end, we represent nodes as independent random variables X_k, following a Bernoulli distribution, with a value $X_k = 1$ if the node is present with probability ϱ, and $X_k = 0$ otherwise (if the node is absent). The probability that an edge (i, j) from $\mathcal{E}_{\max}$ exists in $\mathcal{E}$ is then given by $E[X_i X_j] = \varrho^2$. As such, the expected number of edges in $\mathcal{E}$ follows as $E[\sum_{(i,j)\in\mathcal{E}_{\max}} X_i X_j] = \varrho^2(4L^2 - 6L + 2)$. We have used this relation to generate random ER graphs $\mathcal{G}(n, p)$ with desired (average) edge probability $p = \overline{|\mathcal{E}|}/[n(n-1)/2]$; we note that the variance of $|\mathcal{E}|$ differs between the ER graphs and the UD graphs.

2. Classical hardware

In this section, we specify the classical hardware on which our numerical experiments were run. All B&B-based results obtained with optimization by CPLEX 20.1 were collected from executions of this software using the Python package DOCPLEX with default parameters. The hardware employed is an Intel(R) Xeon(R) Platinum 8259CL CPU @ 2.50GHz with 8 cores. For numerical experiments shown in Fig. 4(b), the median overhead due to multithreading was found to be nearly constant over the entire range of problem sizes. As such, the O complexity is insensitive to the maximum number of threads used by CPLEX. The results for the sweeping line algorithm and simulated annealing, where time-to-solution was considered, were performed on an AMD(R) Ryzen(R) 9 5950X @ 4.90 GHz. Various hardware was employed in sampling instances for hardness and P_{MIS} (where the runtime is irrelevant).

3. Plots description

If not stated otherwise, boxes in box plots correspond to the 16% and 84% percentiles of the $\log_{10}$ variable plotted, and the whiskers refer to the 2% and 98% percentiles. The horizontal line within the box denotes the median value. Points outside the whiskers are plotted individually. This was done to highlight the top 2%, which are used when analyzing the scaling behavior for hard instances.

4. Scaling of TTS with hardness

In this section, we derive the expected scaling of TTS_{99} with the hardness parameter $\mathbb{H}$ for SA. In the main text, for a fixed depth we have provided fits assuming the functional form $P_{\mathrm{MIS}} = 1 - \exp(-C\mathbb{H}^{-\alpha})$; see Fig. 7. Here, we now focus on hard instances where $\mathbb{H}$ is large and P_{MIS} is small. For sufficiently large values of $\mathbb{H}$, we can approximate $P_{\mathrm{MIS}} \approx C\mathbb{H}^{-\alpha}$, giving $P_{\mathrm{MIS}} \sim \mathbb{H}^{-\alpha}$. For small values of P_{MIS}

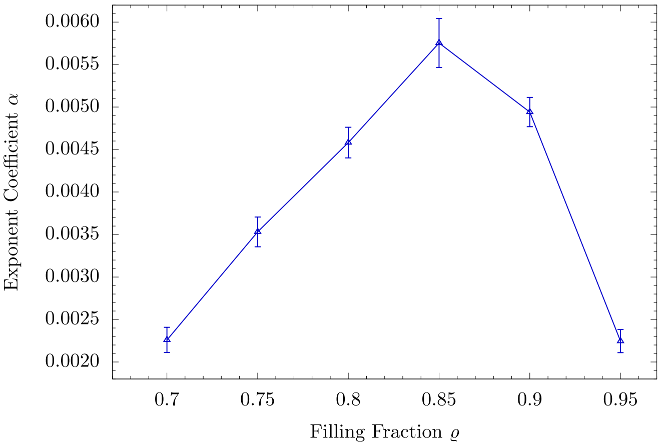

FIG. 10. Coefficient α extracted from time-to-solution (TTS) scaling with system size as TTS $= O(2^{\alpha N})$ for the B&B solver as a function of the lattice filling ϱ. In the main text, we have focused on $\varrho = 0.8$. All results have been taken from the top 2% TTS instances over 1000 random instances. Error bars correspond to the 95% confidence interval.

we have $\mathrm{TTS}_{99} \sim 1/P_{\mathrm{MIS}}$, yielding the scaling

$$\mathrm{TTS}_{99} \sim \mathbb{H}^{\alpha}. \tag{A1}$$

Thus, for large hardness we expect TTS to scale with the same exponent α as found from the fit shown in Fig. 7. We have further verified this result numerically by excluding small systems with small hardness.

APPENDIX B: ADDITIONAL NUMERICAL RESULTS

In this Appendix, we present additional numerical results, complementing the results shown in the main text. In Appendix B 1 we provide additional results for problem hardness as a function of the filling fraction ϱ. In Appendix B 2 we analyze a potential correlation between the hardness parameter $\mathbb{H}$ and TTS for our B&B solvers. In Appendix B 3 we provide additional results for problem hardness as a function of the unit-disk radius r.

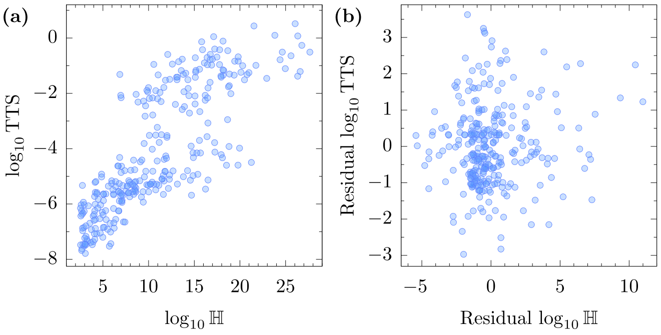

FIG. 11. Correlation analysis for B&B solver. Results are shown for the hardest 2% according to the hardness parameter $\mathbb{H}$ on 1000 instances for each (odd) problem size from $L = 7$ to 35. (a) Scatter plot of hardness $\mathbb{H}$ and TTS for B&B solver. The corresponding Pearson correlation amounts to 0.81. (b) Scatter plot of the residuals from the linear regression of $\mathbb{H}$ and TTS with system size N. Here the Pearson correlation is found to be 0.01.

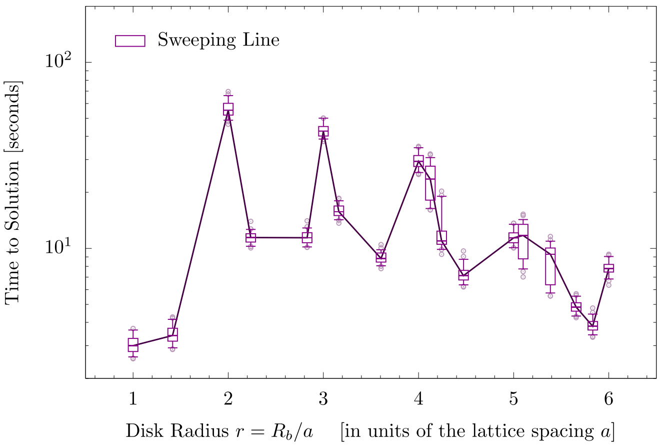

FIG. 12. Time-to-solution (TTS) for the SLA solver as a function of the disk radius r for random UD instances, for system size $L = 21$ and density $\varrho = 0.8$ (i.e., $N \approx 350$), with 100 random seeds per radius. Similar to our results for our B&B solver, we observe distinct peaks at $r = 2, 3, 4$.

1. Filling fraction

In the main text we have focused on instances with Union-Jack-like connectivity and filling fraction $\varrho = 80\%$, following Ref. [12]. In Fig. 10 we provide additional results for problem hardness as a function of the density $\varrho \in [0.7, 0.9]$, given in terms of the coefficient α extracted from TTS scaling with system size as TTS $= \mathcal{O}(2^{\alpha N})$ for the B&B solver. We find that instances with $\varrho \approx 80\%$ are among the hardest instances, with the hardness maximum around $\varrho \approx 85\%$.

2. Correlation of hardness with TTS for B&B solvers

In this section, we analyze a potential correlation between the hardness parameter $\mathbb{H}$ and TTS for our B&B solvers.

At first, we observe a correlation between the logarithms of $\mathbb{H}$ and TTS with a Pearson correlation reaching 0.48. However, when removing a linear dependence with the system size N to both of these quantities, the correlation is found to drop to

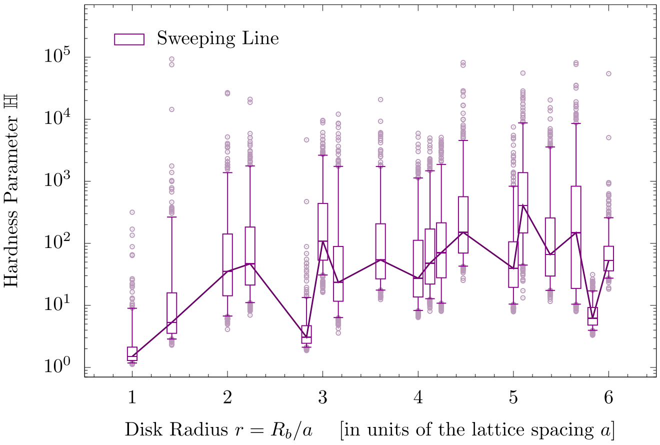

FIG. 13. Hardness parameter $\mathbb{H}$ as a function of the radius of the unit-disk graph (UDG), for system size $L = 21$ and density $\varrho = 0.8$ (i.e., $N \approx 350$), with 1000 random seeds per radius.

0.044. Selecting only the hardest instances (as given by the top 2% of hardness $\mathbb{H}$) exacerbates this phenomenon, as shown in Fig. 11. Note that the linear regression is done without a minimum size threshold because the hardness parameter $\mathbb{H}$ becomes rapidly prohibitively long to evaluate.

3. Additional results for large-radius instances

In this section, we provide additional results for problem hardness as a function of unit-disk radius r for random UD instances.

First we provide results for the exact SLA solver. Our results for TTS as a function of the disk radius r for random UD instances are shown in Fig. 12. Similar to our results for our B&B solver, we observe distinct peaks at $r = 2, 3, 4$, thus further motivating future experiments with Rydberg atom arrays on these instances.

We also analyze the dependence of the hardness parameter $\mathbb{H}$ [as defined in Eq. (3)] on the unit-disk radius r, knowing that $\mathbb{H}$ largely determines problem hardness and thus algorithmic performance for Markov-chain based algorithms such as SA as well as the hybrid quantum algorithm in Ref. [12]. The results of this analysis are displayed in Fig. 13. While we do observe some dependence, we do not observe very pronounced peaks as seen for the B&B and SLA solvers in Figs. 8 and 12, respectively. Again, this observation motivates future experiments with Rydberg atom arrays, as the likelihood for a potential quantum speedup may be larger on these instances, thus potentially helping to identify new regimes where quantum algorithms can be useful.

[1] T. Wilk, A. Gaëtan, C. Evellin, J. Wolters, Y. Miroshnychenko, P. Grangier, and A. Browaeys, Entanglement of two individual neutral atoms using Rydberg blockade, Phys. Rev. Lett. **104**, 010502 (2010).

[2] M. Saffman, T. G. Walker, and K. Mølmer, Quantum information with Rydberg atoms, Rev. Mod. Phys. **82**, 2313 (2010).

[3] M. Saffman, Quantum computing with atomic qubits and Rydberg interactions: Progress and challenges, J. Phys. B **49**, 202001 (2016).

[4] H. Bernien, S. Schwartz, A. Keesling, H. Levine, A. Omran, H. Pichler, S. Choi, A. S. Zibrov, M. Endres, M. Greiner *et al.*, Probing many-body dynamics on a 51-atom quantum simulator, Nature (London) **551**, 579 (2017).

[5] L. Henriet, L. Beguin, A. Signoles, T. Lahaye, A. Browaeys, G.-O. Reymond, and C. Jurczak, Quantum computing with neutral atoms, Quantum **4**, 327 (2020).

[6] C. S. Adams, J. D. Pritchard, and J. P. Shaffer, Rydberg atom quantum technologies, J. Phys. B **53**, 012002 (2019).

[7] S. Ebadi, T. T. Wang, H. Levine, A. Keesling, G. Semeghini, A. Omran, D. Bluvstein, R. Samajdar, H. Pichler, W. W. Ho *et al.*, Quantum phases of matter on a 256-atom programmable quantum simulator, Nature (London) **595**, 227 (2021).

[8] H. Pichler, S.-T. Wang, L. Zhou, S. Choi, and M. D. Lukin, Quantum optimization for Maximum Independent Set using Rydberg atom arrays, arXiv:1808.10816.

[9] H. Pichler, S.-T. Wang, L. Zhou, S. Choi, and M. D. Lukin, Computational complexity of the Rydberg Blockade in two dimensions, arXiv:1809.04954.

[10] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices, Phys. Rev. X **10**, 021067 (2020).

[11] M. F. Serret, B. Marchand, and T. Ayral, Solving optimization problems with Rydberg analog quantum computers: Realistic requirements for quantum advantage using noisy simulation and classical benchmarks, Phys. Rev. A **102**, 052617 (2020).

[12] S. Ebadi, A. Keesling, M. Cain, T. T. Wang, H. Levine, D. Bluvstein, G. Semeghini, A. Omran, J.-G. Liu, R. Samajdar *et al.*, Quantum optimization of maximum independent set using Rydberg atom arrays, Science **376**, 1209 (2022).

[13] M. Cain, S. Chattopadhyay, J.-G. Liu, R. Samajdar, H. Pichler, and M. D. Lukin, Quantum speedup for combinatorial optimization with flat energy landscapes, arXiv:2306.13123.

[14] B. F. Schiffer, D. S. Wild, N. Maskara, M. Cain, M. D. Lukin, and R. Samajdar, Circumventing superexponential runtimes for hard instances of quantum adiabatic optimization, arXiv:2306.13131.

[15] M. Endres, H. Bernien, A. Keesling, H. Levine, E. R. Anschuetz, A. Krajenbrink, C. Senko, V. Vuletic, M. Greiner, and M. D. Lukin, Atom-by-atom assembly of defect-free one-dimensional cold atom arrays, Science **354**, 1024 (2016).

[16] D. Barredo, V. Lienhard, S. de Léséleuc, T. Lahaye, and A. Browaeys, Synthetic three-dimensional atomic structures assembled atom by atom, Nature (London) **561**, 79 (2018).

[17] H. Labuhn, D. Barredo, S. Ravets, S. de Léséleuc, T. Macrì, T. Lahaye, and A. Browaeys, Tunable two-dimensional arrays of single Rydberg atoms for realizing quantum Ising models, Nature (London) **534**, 667 (2016).

[18] V. Lienhard, S. de Léséleuc, D. Barredo, T. Lahaye, A. Browaeys, M. Schuler, L.-P. Henry, and A. M. Läuchli, Observing the space- and time-dependent growth of correlations in dynamically tuned synthetic Ising models with antiferromagnetic interactions, Phys. Rev. X **8**, 021070 (2018).

[19] E. Guardado-Sanchez, P. T. Brown, D. Mitra, T. Devakul, D. A. Huse, P. Schauß, and W. S. Bakr, Probing the quench dynamics of antiferromagnetic correlations in a 2D quantum Ising spin system, Phys. Rev. X **8**, 021069 (2018).

[20] D. Bluvstein, H. Levine, G. Semeghini, T. T. Wang, S. Ebadi, M. Kalinowski, A. Keesling, N. Maskara, H. Pichler, M. Greiner *et al.*, A quantum processor based on coherent transport of entangled atom arrays, Nature (London) **604**, 451 (2022).

[21] M. D. Lukin, M. Fleischhauer, R. Cote, L. M. Duan, D. Jaksch, J. I. Cirac, and P. Zoller, Dipole Blockade and quantum information processing in mesoscopic atomic ensembles, Phys. Rev. Lett. **87**, 037901 (2001).

[22] H. Levine, A. Keesling, G. Semeghini, A. Omran, T. T. Wang, S. Ebadi, H. Bernien, M. Greiner, V. Vuletić, H. Pichler *et al.*, Parallel implementation of high-fidelity multiqubit gates with neutral atoms, Phys. Rev. Lett. **123**, 170503 (2019).

[23] S. J. Evered, D. Bluvstein, M. Kalinowski, S. Ebadi, T. Manovitz, H. Zhou, S. H. Li, A. A. Geim, T. T. Wang, N.

Maskara *et al.*, High-fidelity parallel entangling gates on a neutral atom quantum computer Nature (London) **622**, 268 (2023).

[24] W. K. Hale, Frequency assignment: Theory and applications, Proc. IEEE **68**, 1497 (1980).

[25] Y. Dong, A. V. Goldberg, A. Noe, N. Parotsidis, M. G. C. Resende, and Q. Spaen, A metaheuristic algorithm for large Maximum Weight Independent Set problems, arXiv:2203.15805.

[26] V. Boginski, S. Butenko, and P. M. Pardalos, Statistical analysis of financial networks, Comput. Stat. Data Anal. **48**, 431 (2005).

[27] A. Kalra, F. Qureshi, and M. Tisi, Portfolio asset identification using graph algorithms on a Quantum Annealer, SSRN (2018), doi:10.2139/ssrn.3333537.

[28] J. Wurtz, P. L. S. Lopes, N. Gemelke, A. Keesling, and S. Wang, Industry applications of neutral-atom quantum computing solving Independent Set problems, arXiv:2205.08500.

[29] M. R. Garey, and D. S. Johnson,*Computers and Intractability; A Guide to the Theory of NP-completeness* (Freeman, New York, 1990).

[30] P. K. Agarwal, M. van Kreveld, and S. Suri, Label placement by maximum independent set in rectangles, Computat. Geom. **11**, 209 (1998).

[31] E. J. van Leeuwen, in *Graph-theoretic Concepts in Computer Science*, edited by D. Kratsch (Springer, Berlin, 2005), pp. 351–361.

[32] B. N. Clark, C. J. Colbourn, and D. S. Johnson, Unit disk graphs, Discr. Math. **86**, 165 (1990).

[33] M. Aramon, G. Rosenberg, E. Valiante, T. Miyazawa, H. Tamura, and H. G. Katzgraber, Physics-inspired optimization for quadratic unconstrained problems using a digital annealer, Front. Phys. **7**, 48 (2019).

[34] M. I. Shamos and D. Hoey, in *17th Annual Symposium on Foundations of Computer Science (sfcs 1976), Houston, TX* (IEEE, Piscataway, NJ, 1976), pp. 208–215.

[35] I. I. Cplex, V20.1: User's Manual for CPLEX, International Business Machines Corporation (2021).

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

[37] W. Wang, J. Machta, and H. G. Katzgraber, Comparing Monte Carlo methods for finding ground states of Ising spin glasses: Population annealing, simulated annealing, and parallel tempering, Phys. Rev. E **92**, 013303 (2015).

[38] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, Equation of state calculations by fast computing machines, J. Chem. Phys. **21**, 1087 (1953).

[39] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Optimization by simulated annealing, Science **220**, 671 (1983).

[40] R. H. Swendsen and J.-S. Wang, Replica Monte Carlo simulation of spin-glasses, Phys. Rev. Lett. **57**, 2607 (1986).

[41] J. J. Moreno, H. G. Katzgraber, and A. K. Hartmann, Finding low-temperature states with Parallel Tempering, Simulated Annealing and simple Monte Carlo, Int. J. Mod. Phys. C **14**, 285 (2003).

[42] D. J. Earl and M. W. Deem, Parallel Tempering: Theory, applications, and new perspectives, Phys. Chem. Chem. Phys. **7**, 3910 (2005).

[43] J.-G. Liu, X. Gao, M. Cain, M. D. Lukin, and S.-T. Wang, Computing solution space properties of combinatorial optimization problems via generic Tensor Networks, SIAM J. Sci. Comput. **45**, A1239 (2023).

[44] M.-T. Nguyen, J.-G. Liu, J. Wurtz, M. D. Lukin, S.-T. Wang, and H. Pichler, Quantum optimization with arbitrary connectivity using Rydberg atom arrays, PRX Quantum **4**, 010316 (2023).

[45] M. Kim, K. Kim, J. Hwang, E.-G. Moon, and J. Ahn, Rydberg quantum wires for maximum independent set problems, Nat. Phys. **18**, 755 (2022).

[46] S. Mandrà, Z. Zhu, W. Wang, A. Perdomo-Ortiz, and H. G. Katzgraber, Strengths and weaknesses of weak-strong cluster problems: A detailed overview of state-of-the-art classical heuristics versus quantum approaches, Phys. Rev. A **94**, 022337 (2016).

[47] github.com/jpmorganchase/hardness-of-mis-on-udg