PHYSICS

Quantum-enhanced greedy combinatorial optimization solver

Maxime Dupont[1]*, Bram Evert[1], Mark J. Hodson[1], Bhuvanesh Sundar[1], Stephen Jeffrey[1], Yuki Yamaguchi[1], Dennis Feng[1], Filip B. Maciejewski[2,3], Stuart Hadfield[2,3], M. Sohaib Alam[2,3], Zhihui Wang[2,3], Shon Grabbe[2], P. Aaron Lott[2,3], Eleanor G. Rieffel[2], Davide Venturelli[2,3], Matthew J. Reagor[1]*

Combinatorial optimization is a broadly attractive area for potential quantum advantage, but no quantum algorithm has yet made the leap. Noise in quantum hardware remains a challenge, and more sophisticated quantum-classical algorithms are required to bolster their performance. Here, we introduce an iterative quantum heuristic optimization algorithm to solve combinatorial optimization problems. The quantum algorithm reduces to a classical greedy algorithm in the presence of strong noise. We implement the quantum algorithm on a programmable superconducting quantum system using up to 72 qubits for solving paradigmatic Sherrington-Kirkpatrick Ising spin glass problems. We find the quantum algorithm systematically outperforms its classical greedy counterpart, signaling a quantum enhancement. Moreover, we observe an absolute performance comparable with a state-of-the-art semidefinite programming method. Classical simulations of the algorithm illustrate that a key challenge to reaching quantum advantage remains improving the quantum device characteristics.

INTRODUCTION

A handful of promising classes of quantum algorithms have been advanced for combinatorial optimization problems, such as quantum adiabatic evolution algorithms (*1*), variational quantum algorithms (*2*–*4*), as well as others (*5*, *6*). In all cases, the problem takes the form of an objective function to extremize, which can be interpreted as an Ising-type Hamiltonian whose ground state is the global extremum of the problem. Solving a generic Ising model is NP-hard (*7*), and it remains open whether quantum computers can indeed provide a practical advantage over classical methods. A tremendous amount of work has been dedicated to quantum annealers, which leverage adiabatic evolution (*8*–*10*), while the focus for gate-based quantum computers has been mainly on parameterized quantum circuits like the quantum approximate optimization algorithm (QAOA) (*2*, *3*, *11*–*15*).

The QAOA has been implemented on several experimental platforms to solve a range of combinatorial optimization problems (*13*–*22*). However, a major challenge in these demonstrations has been the stringent technical requirement of reducing hardware noise to provide good quality solutions that are well-separated from trivial classical approaches such as random sampling. In particular, loading arbitrary graph problems, beyond the native topology of the quantum computer, often demands an additional overhead, increasing noise and lowering the performance of the quantum algorithm. For example, while an early implementation of the QAOA on a superconducting quantum system for solving hardware-native maximum cut problems on 19 qubits already showed performance better than random sampling (*13*), an implementation for solving high-dimensional graph problems beyond the hardware-native topology on 23 qubits found results only as good as random guessing on contemporary devices (*15*), with similar trends for Rydberg atoms and trapped ions (*14*, *16*, *17*, *20*, *23*). Yet, it is unavoidable for quantum computers to tackle more intricate problem instances enroute to solving universal and real-world problems.

To date, noise has made even the most straightforward classical optimization approaches better candidates for solving real-world optimization. For instance, classical greedy algorithms, which iteratively build a solution by making the locally optimal choice at each stage, are intuitive, easy to implement, and will most likely outperform a modern noisy quantum computer. This raises the question: Can one design algorithms using current quantum technologies to their advantage with performance guarantees, making them realistically competitive against classical ones for arbitrary problems at scale?

Overall, our goal is to minimize the objective function

$$C = u + \sum_{i=1}^{N} v_i Z_i + \sum_{j<i}^{N} w_{ij} Z_i Z_j \tag{1}$$

where u, v_i, and w_{ij} are problem-specific scalar parameters, and $Z_i \in \{-1, +1\}$ are Ising spin variables with corresponding bit values $B_i = 1/2 - Z_i/2 \in \{0,1\}$. The goal is to find a bit string $\mathbf{B} = (B_1, B_2, ..., B_N)$ minimizing Eq. 1. Substituting the Ising variables for Pauli Z operators (we use the same symbol for both when clear from context) in Eq. 1, the minimization can be achieved on a quantum computer using quantum adiabatic evolution or variational algorithms. The latter, such as the QAOA [2, 3], are better suited for near-term digital devices with their theoretical performance at least on par with a discretized adiabatic evolution with a fixed number of layers. A QAOA circuit with p layers reads

$$|\gamma, \beta\rangle = [\prod_{d=1}^{p} U_M(\beta_d) U_{PS}(\gamma_d)] H^{\otimes N} |0\rangle^{\otimes N} \tag{2}$$

[1]Rigetti Computing, Berkeley, CA 94710, USA. [2]QuAIL, NASA Ames Research Center, Moffett Field, CA 94035, USA. [3]USRA Research Institute for Advanced Computer Science, Mountain View, CA 94035, USA.
*Corresponding author. Email: mdupont@rigetti.com (M.D.); matt@rigetti.com (M.J.R.)

where H is the one-qubit Hadamard gate. The unitaries U_{PS} and U_{M} are called the phase separator and the mixer, respectively, and are parametrized with the real-valued angles γ_d and β_d, respectively. They are defined as

$$U_{\mathrm{PS}}(\gamma_d) = e^{i\gamma_d C} \text{ and } U_{\mathrm{M}}(\beta_d) = e^{i\beta_d \sum_{j=1}^{N} X_j} \quad (3)$$

where C is the operator corresponding to the objective function of Eq. 1 and where X_j is the Pauli operator on qubit j. The search for optimal angles γ^* and β^* is done in a quantum-classical hybrid fashion by minimizing the expectation value $\langle C\rangle = \langle \gamma, \beta \mid C \mid \gamma, \beta\rangle$. Candidate bit string solutions $\{\mathbf{B}\}$ are obtained by sampling the quantum state $|\gamma^*, \beta^*\rangle$.

The quality of a solution is ranked by its cost value C^* with respect to the maximum (worst) and minimum (optimal) ones, $C_{\max}$ and $C_{\min}$, respectively, through the approximation ratio

$$r = (C_{\max} - C^*)/(C_{\max} - C_{\min}) \quad (4)$$

which is equal to 1 for an optimal solution $C^* = C_{\min}$. On difficult problem instances, it is proven to be NP-hard to achieve an ensemble-average approximation ratio greater than a given value $r^\star$, and intensive ongoing theoretical efforts (*2*, *3*, *11*, *24*–*30*) are attempting to establish whether QAOA with p layers can lead to an average approximation ratio larger than the best-known classical methods on some problem classes. Because of noise, when QAOA is run on current generation (noisy) quantum hardware, it leads to approximation ratios much smaller than theoretical bounds and those obtained on small-scale classical emulations (*13*–*15*). In a strong noise regime where the quantum state tends to be described as a maximally mixed state, the expected performance is that of a random bit string sampling. Under the assumption that the spectrum of C (Eq. 1) is symmetric about u, this corresponds to $r = 1/2$.

In the following, we develop a hybrid classical-quantum algorithm performing as good as a randomized classical greedy algorithm in the presence of strong noise. For instance, the classical greedy approach has a performance $r \simeq 0.848497\ldots$ for Sherrington-Kirkpatrick (SK) Ising spin glass problems (see Methods), much larger than random guessing and a very noisy vanilla QAOA execution. We implement the quantum algorithm on Rigetti Aspen-M-3 programmable superconducting quantum system using up to 72 qubits and find that it systematically outperforms its classical greedy counterpart, signaling a quantum enhancement.

RESULTS

We introduce an iterative algorithm for solving discrete optimization problems which bears similarities with divide-and-conquer methods (*31*) and more closely with other iterative/recursive techniques referred as RQAOA (*32*,*33*) or greedy decompositions in quantum annealing (*34*)—the main difference being the freezing procedure. At each iteration, a set of variables are frozen to their classical values depending on the output returned by a quantum computer—although, as discussed later, the approach works with any sampleable distribution over bit strings, quantum or classical. These variables are removed, and an updated, smaller optimization problem is generated. The procedure is repeated until all variables are frozen or until the remaining problem is small enough for brute force.

A main difference between the quantum-enhanced greedy algorithm that we develop and prior iterative approaches is its robustness to noise, which is key when executing quantum algorithms at scale on current quantum hardware. In a strong depolarizing noise regime, our quantum algorithm maps to a classical randomized greedy algorithm for which one can analytically estimate its average performance for problems such as SK Ising spin glasses. In the same strong noise regime, as expectation values, such as two-point correlations $\langle ZiZj\rangle$, tend towards zero, other existing iterative algorithms (*31*–*34*) would perform as well as a random sampling strategy in the absence of a mapping to a classical greedy baseline.

Each iteration ℓ of our quantum-enhanced greedy algorithm follows the steps,

#1. Obtain a list of $M^{(\ell)}$ bit strings $\{\mathbf{B}^{(\ell)}\}$ that encode candidate solutions to the problem. In the quantum version, these bit strings are sampled from the output of the quantum computer, where the quantum circuit optimizes the objective function $C^{(\ell)}$ (Eq. 1). In the classical randomized greedy version, these bit strings are sampled from a uniform distribution of all bit strings.

#2. Find a set $\{k\}$ of K variables to freeze. Different heuristics can be envisioned (*32*, *33*), including a majority vote based on one-body expectation values, i.e., $\max_i \mid \langle Z_i\rangle|$, or based on two-body expectation values, as well as many others. In the following, we use a two-body expectation strategy developed in Methods to select $K = 1$ variable.

#3. Find the frozen value of each variable selected in #2: For each variable in $\{k\}$, loop over all the possible values $\{s\}$ of the variable (i.e., $s = 0$ or 1 for a binary variable), substitute them for all bit string of the list such that $\{\mathbf{B}_k^{(\ell)} \leftarrow s\}$, and compute the expectation value of the cost based on the modified $\{\mathbf{B}^{(\ell)}\}$. The assignment s leading to the best expectation value of the cost is taken as the variable's frozen value (see Methods).

#4. Update the problem $C^{(\ell)}$ by replacing each of the operators in $\{Z_k\}$ by a constant based on the expectation value of $\langle Z_k\rangle$. The scalars are absorbed into u, v_i, and w_{ij} in Eq. 1 to create a new problem $C^{(\ell+1)}$ with at least K fewer variables (see Methods).

Other strategies can be implemented in the third step (#3). The essential point is that for random bit strings $\{\mathbf{B}^{(\ell)}\}$, the freezing decisions are locally optimal with respect to the objective function, independently of the selected variables. As such, the average approximation ratio from random bit strings and optimal freezing is $r \simeq 0.848497\cdots$ (see Methods). Also, if $\{\mathbf{B}^{(\ell)}\}$ is replaced by optimal bit strings with respect to the objective function, then the above algorithm will preserve an optimal solution, and yield $r = 1$. Therefore, the intuition is that for bit strings which are between random and optimal, there should be a performance boost with respect to the classical greedy baseline. Better-than-random bit strings, on average, should help make better-informed decisions for the selection and thus guide an otherwise randomized greedy process. The complexity of the above algorithm is $O[(N/K)N_{\mathrm{edges}}]$ with N_{edges} the number of two-body terms in the graph problem. Taking $K \sim O(1)$ and $N_{\mathrm{edges}} \sim O(N^2)$ for the SK instances considered in the following leads to $O(N^3)$ complexity. We note that the complexity of the classical randomized version of the algorithm can be reduced to $O(N^2)$ when not working with explicit bit strings and considering that all expectation values average to zero.

We implement the quantum-enhanced greedy algorithm on SK problem instances (*35*) by setting $u = v_i = 0$ in Eq. 1 and draw the parameters w_{ij} uniformly from $\{+1, -1\}$. SK models correspond to paradigmatic Ising spin glasses. Although it was recently proven that the ground state energy of SK models can be efficiently approximated with an approximation ratio $(1 - \varepsilon)$ by an approximate message passing algorithm (*36*), SK models remain a relevant benchmark for combinatorial optimization methods.

We run the algorithm on Rigetti's superconducting quantum processor Aspen-M-3 with a planar square-octagon topology of 79 qubits (see the Methods for the parameters used in practice and more details). The limited connectivity of the hardware, displayed in Fig. 1, requires an extensive swap network to cover two-qubit gates between arbitrary qubits (*37*). Consequently, implementation of the phase separator unitary of Eq. 2 is not practical (due to noise) for SK problems with large N (Fig. 1, A and B]. Instead, we use a truncated one-layer QAOA ansatz: At each iteration of the algorithm, the problem is randomly mapped to the hardware-native architecture, and only gates involving qubits connected within 2 swap cycles are considered, with the others dismissed, as exemplified in Fig. 1C. The circuit (Eq. 2) is compiled into hardware-native gates, resulting in about 400 native $\sqrt{\text{iSWAP}}$ two-qubit gates for the largest problems considered, as detailed in Methods. We collect a total of $M^{(\ell)} = 256$ bit strings for all steps ℓ.

We use the quantum-enhanced greedy algorithm to solve a set of 10 random SK problem instances for sizes $N = 8$, 24, 40, 56, and 72. We freeze one variable ($K = 1$) at a time and use two-body expectation values to inform the selection process. We report the estimated expectation value of the approximation ratio $r \equiv \langle r \rangle_{\gamma^*, \beta^*}$ by computing the expectation value of the corresponding objective function over all sampled candidate bit string solutions at angles γ^* and β^*. For $N \leq 24$, we use brute force to compute $C_{\max}$ and $C_{\min}$ (Eq. 4) for a given problem instance. For larger N, we rely on the fact that the cost of the optimal solution is self-averaging, known exactly for $N \to +\infty$, and that finite-size corrections have also been studied over an ensemble of random instances (*38*). This gives access to a proxy for approximating r, assuming one is not interested in the performance for an individual problem but that of an ensemble. Additional detail is given in Methods.

We show the obtained approximation ratio as a function of the iteration step in Fig. 2. Iteration step 0 corresponds to a truncated one-layer QAOA ansatz run on the initial problem of N variables. For all sizes, this is slightly above the $r = 1/2$ random sampling bar, emphasizing that the QAOA displays a low average performance on current hardware. The last step corresponds to the final solution of the quantum-enhanced greedy algorithm. Its average performance is systematically above that of the classical greedy baseline. We note that the approximation ratio in both the classical greedy and the quantum-enhanced algorithms have a distribution around the average. This is more evident at small N, for example, $N = 8$ and $N = 24$ in Fig. 2, where the quantum greedy results for some problem instances dip below the average classical greedy results. The estimated approximation ratio at the last step is displayed in Fig. 3 as a function of problem size N. We observe a decrease in performance with increasing problem size, which has two primary causes. First, the larger the size, the larger the quantum circuit, leading to a higher error rate. Second, the phase separator unitary of the truncated QAOA ansatz only covers an $O(1/N)$ density of edges of the graph problem, which accounts for a vanishing fraction of two-body terms in the SK problem instances (about 3% for $N = 72$). For comparison, a noiseless simulation of a nontruncated standard single-layer QAOA circuit leads to an average approximation

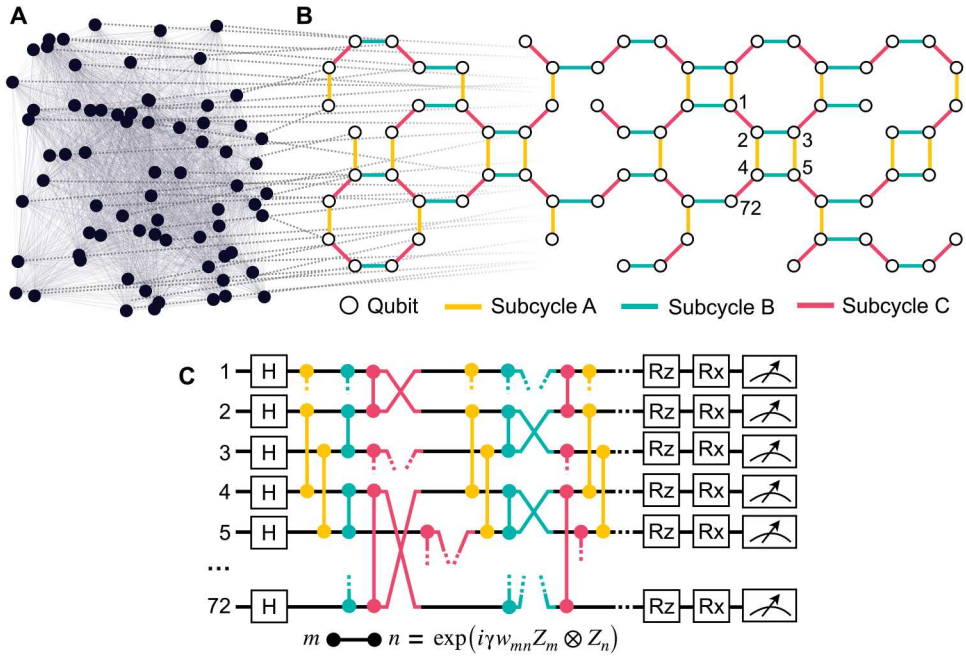

Fig. 1. Problem mapping and quantum circuit. (**A**) Binary optimization problem with $N = 72$ variables (Eq. 1). (**B**) Binary variables are randomly mapped to a qubit of the square-octagon topology of the Rigetti Aspen-M-3 quantum processor. The edges are divided into three independent sets *A*, *B*, and *C* on which two-qubit gates can be executed in parallel. (**C**) QAOA quantum circuit (Eq. 2). We show a truncated version run on hardware: We apply R_{zz} gates on the yellow edges in subcycle A, then R_{zz} gates on the teal edges in subcycle B, then R_{zz} and SWAP gates on the magenta edges in subcycle C, and so on as shown above.

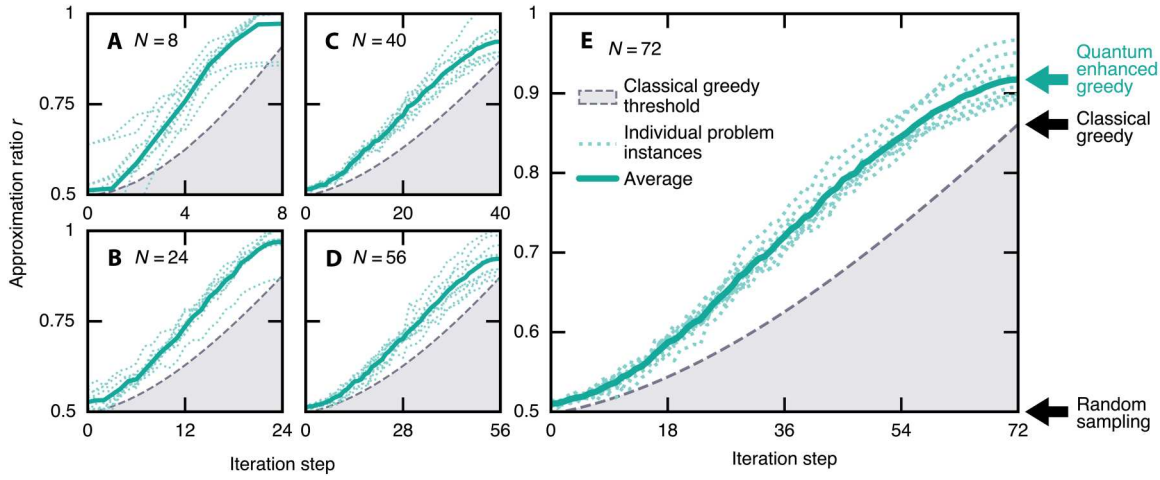

Fig. 2. Approximation ratio versus iteration step. Each panel corresponds to an SK problem instance of different size: (**A**) $N = 8$, (**B**) $N = 24$, (**C**) $N = 40$, (**D**) $N = 56$, and (**E**) $N = 72$. A random sampling strategy leads to an average approximation ratio $r = 0.5$. The performance of the classical greedy baseline is shown by the shaded region with a final average approximation ratio $r \simeq 0.848497\ldots$ for $N \to +\infty$ (see Methods). The quantum-enhanced greedy data show the expectation value of the approximation ratio for 10 randomly generated problem instances (see Fig. 1). The performance at iteration step 0 is that of the truncated one-layer QAOA. The performance at iteration step N is that of the quantum-enhanced algorithm, also reported in Fig. 3. Because we display all individual instances, we omit the error bar for the average case.

ratio $r = 1/2 + 1/4P\sqrt{e} \simeq 0.698688\ldots$ on large SK problem instances (*29*), where P is the Parisi constant (*39*, *40*).

Embedding the QAOA into the quantum-enhanced algorithm greatly enhances the quality of the end result. From step 0 to step N of the iterative loop (Fig. 2), we experimentally obtain a six- to sevenfold increase in the average approximation ratio for $N = 72$ where $1 - r \simeq 0.49 \to 0.08$ (Fig. 3). A fairer comparison point is the classical greedy baseline which runs the same algorithm with random bit strings as input instead. For the SK problem instances considered here, the expected approximation ratio of this classical heuristic is $r \simeq 0.848497\ldots$ as $N \to +\infty$ (see Methods). It is a high absolute bar to pass, much higher than what we obtain from a noisy and truncated one-layer QAOA run at iteration step 0. For reference, it requires at least a perfectly executed nontruncated four- to five-layer QAOA circuit to meet this classical performance [see the Supplementary Materials and (*29*)]. Here, the quantum-enhanced greedy algorithm run on noisy quantum hardware improves upon the average approximation ratio of its classical counterpart by about a factor 2 ($1 - r \simeq 0.151503\ldots \to 0.08$) for the largest problem size $N = 72$ (Fig. 3), empirically confirming the intuition that better-than-random bit strings should, on average, help make better-informed decisions in the freezing process. An explanation is that while bit strings look close to random as a whole because of noise, they might still locally retain relevant information. Here, we look for the most correlated variable (see Methods), as this suggests a well-defined value for the corresponding bit, making it a good candidate for freezing. We observe an absolute performance comparable with state-of-the-art semidefinite programming (SDP) method, corresponding to a spectral relaxation rounding to ±1 each entry of the leading eigenvector of the adjacency matrix of the graph problem (*41*–*43*).

DISCUSSION

While our empirical results demonstrate that the quantum-enhanced algorithm can outperform the classical greedy threshold, signaling a quantum boost, proving this rigorously remains as future work. All the iterative algorithm needs is a bit string generator at step #1, analogous to classical methods such as genetic algorithms and Monte Carlo methods. We highlight the work ahead for reaching a quantum advantage by using noiseless classical tensor network simulations to generate bit strings from a truncated one-layer QAOA circuit with two swap cycles embedded into a one-dimensional lattice (see Methods). This leads to an average approximation ratio $r \simeq 0.95$ for the largest problems considered (Fig. 3), higher than what we obtained with the quantum runs. We, therefore, expect that the performance of the quantum-enhanced greedy algorithm will continue to improve with advances in hardware fidelity. The algorithm can use expectation-based error mitigation techniques (*44*) for improving the freezing decisions—and therefore, the final solution—, which are otherwise not usable for more traditional quantum optimization methods when one is typically interested in enhancing an individual bit string. Further, classical postprocessing methods (*45*) may be leveraged to develop and explore more sophisticated variable freezing procedures, such as postselecting only bit strings that do at least as well as random guessing. Ultimately, the backbone remains the quantum device, and the overarching goal should be to improve its characteristics. Improving those will allow running deeper QAOA circuits as well as other state-of-the-art quantum circuits, leading to an overall performance increase through better-informed decisions (see the Supplementary Materials). This would warrant additional, comprehensive benchmarks against a panoply of state-of-the-art classical algorithms and techniques, such as Ising machines (*46*), to assess on the existence of a practical quantum speedup or advantage.

The freezing decisions have a classical component that can be adapted to deal with some hard constraints; such as postselecting on valid solutions. Hard constraints are ubiquitous in real-world optimization problems and are notoriously difficult to handle in practice. A possible strategy is to design quantum circuits working within the in-constraint space with dynamics restricted to the subspace of feasible solutions (*12*, *47*, *48*), but these

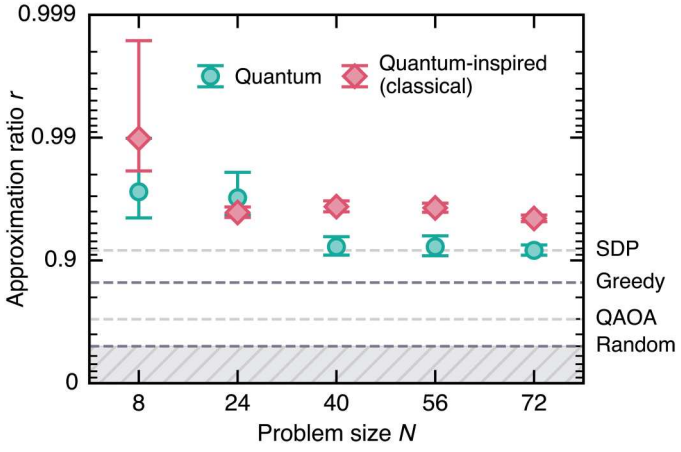

Fig. 3. Performance boost. Approximation ratio based on different bit string generators: The hardware-run QAOA of Fig. 2, averaged over 10 problem instances, and quantum-inspired classical tensor network simulations (see Methods), averaged over 100 random SK problem instances. The different horizontal dashed lines correspond to the average performance of various algorithms in the limit $N \to +\infty$. From bottom to top: Random sampling $r = 1/2$, a perfectly executed single-layer, nontruncated, QAOA circuit $r = 1/2 + 1/4P\sqrt{e} \simeq 0.698688\ldots$ (*29*), the classical greedy baseline $r = 1/2 + \sqrt{2/\pi}/3P \simeq 0.848497\ldots$ (see Methods), a classical SDP approach $r = 1/2 + 1/\pi P \simeq 0.917090\ldots$ (*41*–*43*) where P is the Parisi constant (*39*, *40*). Error bars indicate 1 standard deviation. Numerical values are tabulated in the Supplementary Materials.

methods may require greater quantum resources. Another typical approach uses penalty terms that will disfavor the appearance of out-of-constraint bit strings, but implementing them on near-term devices and tuning their strength can be similarly challenging. Here, the idea is to make only freezing decisions which do not violate any constraints. Note that this is only possible for some classes of constraints, as a general satisfiability problem is NP-complete. We investigate a proof of concept of the modified algorithm on a constrained binary portfolio optimization problem in the Supplementary Materials, comparing it to the state of the art (*49*).

In practice, scaling to hundreds of qubits will require freezing variables simultaneously to keep the runtime under control. We suggest the following modification (*32*): After selecting K variables in step #2, the decision in step #3 attempts the 2^K substitutions in $\{\mathbf{B}^{(\ell)}\}$ and keeps the best one.

Last, it is interesting to think of iterative hybrid classical-quantum setups with performance guarantees on noisy hardware in the context of quantum problems rather than classical ones. For instance, can one enhance the performance of variational quantum eigensolvers (*4*, *50*) for chemistry and other quantum many-body problems by embedding ideas from real-space renormalization group methods, such as linked-cluster expansions or the contractor renormalization group technique (*51*)?

METHODS

Rigetti Aspen-M-3 superconducting platform

Rigetti's Aspen-M-3 is a programmable and universal superconducting quantum computer based on transmon qubits. There are 79 qubits arranged on a planar square-octagon topology. We make use of one-qubit rotation gates about the x axis $Rx(\phi \in \mathbb{Z}) = \exp(-iX\phi\pi/4)$, one-qubit rotation gates about the z axis $Rz(\theta \in \mathbb{R}) = \exp(-iZ\theta/2)$, and the two-qubit gate $\sqrt{\text{iSWAP}} = \exp[i\pi(XX + YY)/8]$ (*52*). X, Y, and Z are Pauli operators. The qubits have an average relaxation time $T_1 = 25(2)$ μs, an average dephasing time $T_2 = 28(2)$ μs, an average readout fidelity of 94.6(7)%, and an average one-qubit Rx fidelity of 99.4(2)% estimated by randomized benchmarking (*53*).

To estimate the fidelity of the operations $\sqrt{\text{iSWAP}}$, we use cycle benchmarking (*54*). In the circuit, cycles were constructed by splitting the three independent edge groups of Fig. 1C each into two clock cycles (for a total of six), where the $\sqrt{\text{iSWAP}}$ gates are separated by at least one idle qubit. This is done to minimize cross-talk between the $\sqrt{\text{iSWAP}}$ gates. Cycle benchmarking provides a measurement of fidelity that is less forgiving, but more realistic, than isolated randomized benchmarking. The cycles are benchmarked the same way they are played in the application circuit, accumulating additional error from decoherence while idling and from the single-qubit gates which precede each two-qubit gate in the cycle. We report in Fig. 4 the marginal process fidelity contributed by each pair of qubits playing an entangling gate in the cycle. In general, the majority of the infidelity is contributed by the entangling operations. We expect further refinements to the cycle calibration process to reduce control errors and yield better cycle performance.

As the quantum-enhanced greedy algorithm iteratively reduces the size of the problem and thus the number of qubits at each iteration step, we successively target smaller and smaller subsets of qubits of higher and higher overall quality. This is visible from the general trends of Fig. 4 where the mean and median fidelities increase and the spread decreases with each iteration step.

Hardware-native quantum circuits

The unitaries of the QAOA in Eq. 3 are expressed through one- and two-qubit gates exclusively, as pictured in Fig. 1C. These gates are further decomposed into the hardware native gate set. For instance, the one-qubit gate Rx(ϕ) is implemented for arbitrary angles using the standard "ZXZXZ" decomposition (*55*). We use the hardware-

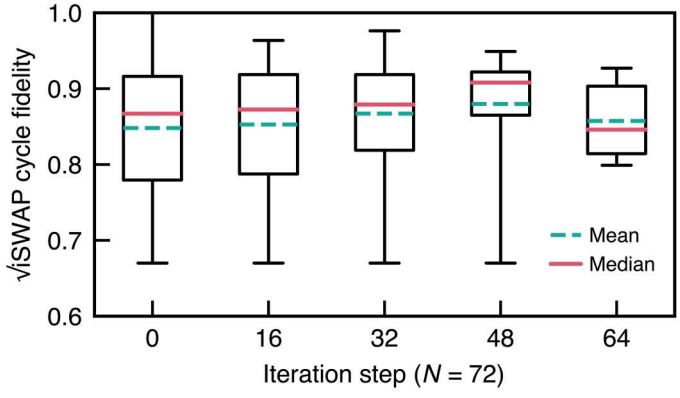

Fig. 4. Cycle fidelity for two-qubit gates. Statistics for the cycle fidelity of $\sqrt{\text{iSWAP}}$ at different iteration steps for a problem size $N = 72$. The boxes cover the first to the third quartile. The whiskers cover the whole range of the data. With each iteration step of the algorithm, the size of the problem is reduced and we successively target smaller and smaller subsets of qubits of higher and higher overall quality resulting in an increasing mean and median with decreasing spread.

native two-qubit $\sqrt{\text{iSWAP}}$ gate to express Rzz(ϕ) = exp (− $i\phi ZZ/2$) and SWAP. Because we always precede a SWAP gate by Rzz(ϕ), we decompose directly the combination of these two gates. Both Rzz and Rzz × SWAP can be implemented using hardware-native one-qubit gates and at most two and three two-qubit $\sqrt{\text{iSWAP}}$ gates (*56*), respectively, as shown in Fig. 5; the precise decomposition is given in the Supplementary Materials. While this may not be the optimal synthesis in general, it is optimal in terms of number of pulses for the Rigetti Aspen-M-3 quantum processor with respect to alternatives.

Optimization of QAOA circuits and sampling bit strings

A QAOA circuit with p layers is defined in Eq. 2. It is parameterized by $2p$ angles, which should be such that the resulting quantum state minimizes the expectation value of the objective function C of Eq. 1. In this work, we consider truncated one-layer QAOA circuits (Fig. 1B) with only two angles γ and β. As such, we optimize them through a 16 × 16 grid search in the range γ ∈ [0,2π] and β ∈ [0, π]. For each of the 16 × 16 = 256 pairs of angles, we collect M = 256 bit strings {**B**} by sampling the quantum state |γ, β⟩ to compute the expectation value

$$\langle C\rangle_{\gamma,\beta} = \langle \gamma,\beta \mid C \mid \gamma,\beta\rangle = \frac{1}{M}\sum\nolimits_{\{\mathbf{B}\}}\langle \mathbf{B} \mid C \mid \mathbf{B}\rangle \quad (5)$$

The angles leading to the minimum expectation value of $\langle C\rangle_{\gamma,\beta}$ are γ^* and β^*. The M bit strings that led to the minimum expectation value $\langle C\rangle_{\gamma^*,\beta^*}$ during the grid search are used in the first step (#1) of the iterative quantum-enhanced greedy algorithm. Examples of $\langle C\rangle_{\gamma,\beta}$ are shown in the Supplementary Materials.

Iterative process

We provide additional detail on our implementation of the iterative steps #2, #3, and #4 of the algorithm. At this stage, we have already obtained a list of bit strings {**B**} sampled from the quantum state of the optimized QAOA circuit at angles γ^* and β^*. At step #2, we find K = 1 variable to freeze based on a two-body expectation strategy. Precisely, for each active node k of the graph, we compute

$$F_k = \frac{1}{M}\Big[\sum\nolimits_{i\in\{\text{active}\}\neq k} \mid w_{ik}\sum\nolimits_{\{\mathbf{B}\}}\langle \mathbf{B} \mid Z_iZ_k \mid \mathbf{B}\rangle \mid + \mid \nu_k\sum\nolimits_{\{\mathbf{B}\}}\langle \mathbf{B} \mid Z_k \mid \mathbf{B}\rangle \mid\Big] \quad (6)$$

where we find k such that $\max_k F_k$. The function F_k corresponds to the expectation value of the objective function at the given iteration step where all the terms involving the node k have been individually sandwiched by a modulus symbol. We now move to step #3 and generate two modified versions of the bit strings {**B**} by setting the bit k on all the M bit strings to either 0 or 1. We then compute the expectation value of the cost $\langle C_0\rangle = \sum_{\{\mathbf{B}_k \leftarrow 0\}} \langle \mathbf{B} \mid C \mid \mathbf{B}\rangle/M$ and $\langle C_1\rangle = \sum_{\{\mathbf{B}_k \leftarrow 1\}} \langle \mathbf{B} \mid C \mid \mathbf{B}\rangle/M$. The value B_k = 0 or 1 which provided the smallest of $\langle C_0\rangle$ and $\langle C_1\rangle$ will be used as the variable's frozen value $\sigma_k = (-1)^{B_k}$. Following the freezing decision of the variable k, the problem is updated as follows

$$\begin{gathered} w_{ik}Z_iZ_k \rightarrow \nu\prime_i Z_i \forall i \text{ with } \nu\prime_i = \nu_i + w_{ik}\sigma_k, \\ \nu_k Z_k \rightarrow u' = u + \nu_k\sigma_k \end{gathered} \quad (7)$$

where the notation of Eq. 1 is used for scalar parameters u, v_i, and w_{ij}.

Classical simulations of quantum circuits

We supplement the quantum experimental results with classical simulations of the circuits. We use two methods throughout this work. The first one, used in the Supplementary Materials, is based on a state vector approach where the quantum state for N qubits is represented as a complex vector of 2^N components. It is an exact method. The second one, used for generating the data of Fig. 3, is a tensor network approach based on matrix product states (*57*). The circuits we simulate with matrix product states are a truncated one-layer QAOA circuit with two swap cycles embedded into a one-dimensional lattice with open boundary conditions. The circuits are shallow enough to be executed exactly with a relatively low bond dimension (the bond dimension is a control parameter of a matrix product state simulator), independent of the number of qubits involved, thus enabling exact classical simulations at N = 72. Precisely, the circuits have a brick wall pattern of two-qubit gates with a total of four layers in addition to one-qubit gates. Additional details on matrix product states are given in the Supplementary Materials.

Optimal cost of SK instances

Computing the approximation ratio of Eq. 4 requires the extremum (best $C_{\min}$ and worst $C_{\max}$) cost values for the SK problem instance of interest. For problems $N \leq 24$, we compute them through brute force by enumerating the 2^N bit strings. For larger N, we rely on the fact that the cost of the optimal solution is self-averaging, known exactly for $N \rightarrow +\infty$, and that finite-size corrections have also been studied over an ensemble of random instances (*38*). In particular, one has,

$$C_{\min} \times N^{-3/2} \simeq -\mathrm{P} + aN^{-\omega} \quad (8)$$

where P = 0.763166726566547... is a universal constant known as the Parisi value (*39*, *40*), ω = 2/3 is a universal exponent accounting for finite-N corrections, and $a \simeq 0.70(1)$ is a nonuniversal constant (*38*). Moreover, on average, $C_{\max} = -C_{\min}$ since it corresponds to solving an equivalent problem with flipped signs for the parameters

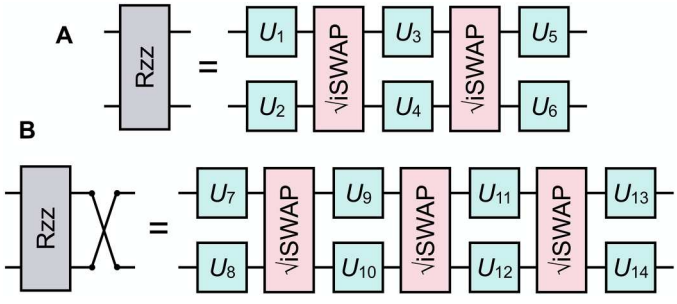

Fig. 5. Two-qubit gate decompositions. (**A**) Decomposition of the parametric two-qubit gate Rzz(ϕ) = exp (− $i\phi ZZ/2$) using two hardware-native $\sqrt{\text{iSWAP}} = \exp[i\pi(XX+YY)/8]$ gates. *X*, *Y*, and *Z* are Pauli operators. (**B**) Decomposition of the parametric two-qubit gate Rzz(ϕ) directly followed by a SWAP using three hardware-native $\sqrt{\text{iSWAP}}$ gates. In each case, the one-qubit gates $U_{1\ldots14}$ carry the angle ϕ and are decomposed explicitly as a function of *Rx* and Rz gates (see the Supplementary Materials for details).

w_{ij} (Eq. 1) (see the Supplementary Materials). Hence, for an ensemble of SK problems leading to an average cost value of C^*, we approximate the obtained approximation ratio as

$$r \simeq \frac{1}{2}\left(1 + \frac{C^*}{C_{\min}}\right) \tag{9}$$

where $C_{\min}$ is evaluated numerically using Eq. 8. For randomly generated bit strings, C^* is symmetrically distributed between $C_{\min}$ and $C_{\max}$, i.e., $C^* = 0$ in expectation, and the corresponding average approximation ratio is $r \simeq 1/2$.

Classical greedy baseline

The classical greedy algorithm iteratively builds a solution by making the locally optimal choice at each stage. There are as many iteration steps as variables in the problem. We note $Z_i = \pm 1$ the contribution to the cost function of the frozen variable i. We consider the SK problem instances of the main text (Eq. 1 with $w_{ij} = \pm 1$, $u = v_i = 0$) for which the freezing decisions go as follows. Iteration step 1, freeze Z_1 arbitrarily to ±1. Iteration step 2, freeze Z_2 such that $w_{12}Z_1Z_2$ is minimized, i.e., $\min_{Z_2}[w_{12}Z_1Z_2]$. Iteration step 3, freeze Z_3 such that $\min_{Z_3}[Z_3(w_{13}Z_1 + w_{23}Z_2)]$. At iteration step ℓ, freeze Z_ℓ such that $Z_\ell\sum_{i=1}^{\ell-1} w_{i\ell}Z_i$ is minimized. This is repeated until iteration step N. The final cost value is

$$C_{\text{greedy}} = -\sum_{\ell=2}^{N} \left| \sum_{i=1}^{\ell-1} w_{i\ell}Z_i \right| \tag{10}$$

The absolute value terms in Eq. 10 can be seen as individual random walks containing $\ell - 1$ steps of length ±1. As such, for $\ell \to +\infty$, their average contribution will be $\simeq \sqrt{2(\ell-1)/\pi}$. This step is analogous to averaging over different freezing orders for the variables. Under the hood, this corresponds to running the algorithm of the main text with a finite number of random bit strings from which statistical fluctuations will lead to different freezing selections in #2 from one run to the next. Hence, one finds that, on average,

$$C_{\text{greedy}} \simeq -\sqrt{2/\pi}\sum_{\ell=1}^{N-1}\sqrt{\ell} \tag{11}$$

As $N \to +\infty$, an asymptotic expansion based on Euler-Maclaurin formula leads to $\sum_{\ell=1}^{N-1}\sqrt{\ell} \sim (2/3)N^{3/2} + \ldots$,, which makes it possible to evaluate the average approximation ratio of the classical greedy algorithm using Eqs. 8 and 9 for an infinite-size SK problem instance

$$\frac{C_{\text{greedy}}}{C_{\min}} = \frac{2\sqrt{2/\pi}}{3\text{P}} \Rightarrow r = \frac{1}{2} + \frac{\sqrt{2/\pi}}{3\text{P}} \simeq 0.848497\ldots \tag{12}$$

The case of finite N is studied numerically in the Supplementary Materials. By definition, this is also the average approximation ratio of the quantum-enhanced algorithm with random input. Physically, this would mean that the quantum computer generates a maximally mixed state $\rho = I/2^N$, as the result of, e.g., strong depolarizing noise: Here, ρ is the density matrix describing N qubits and I the Identity matrix of dimensions $2^N \times 2^N$.

For comparison, other algorithms yield the following average performance: Random sampling $r = 1/2$, a perfectly executed single-layer, nontruncated, QAOA circuit $r = 1/2 + 1/4\text{P}\sqrt{e} \simeq 0.698688\ldots$ (*29*), a classical SDP approach $r = 1/2 + 1/\pi\text{P} \simeq 0.917090\ldots$ (*41*–*43*). We extend nonexhaustively the analytical analysis of the classical greedy baseline to other problems in the Supplementary Materials.

Supplementary Materials

This PDF file includes:

Supplementary text
Figs. S1 to S11
Table S1
References

REFERENCES AND NOTE

1. E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, D. Preda, A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. *Science* **292**, 472–475 (2001).
2. E. Farhi, J. Goldstone, S. Gutmann. A quantum approximate optimization algorithm. arXiv :1411.4028 [quant-ph] (2014).
3. E. Farhi, J. Goldstone, S. Gutmann. A quantum approximate optimization algorithm applied to a bounded occurrence constraint problem. arXiv :1412.6062 [quant-ph] (2014).
4. M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, P. J. Coles, Variational quantum algorithms. *Nat. Rev. Phys.* **3**, 625–644 (2021).
5. A. Montanaro, Quantum speedup of branch-and-bound algorithms. *Phys. Rev. Res.* **2**, 013056 (2020).
6. C.-M. Alexandru, E. Bridgett-Tomkinson, N. Linden, J. MacManus, A. Montanaro, H. Morris, Quantum speedups of some general-purpose numerical optimisation algorithms. *Quantum Sci. Technol.* **5**, 045014 (2020).
7. F. Barahona, On the computational complexity of Ising spin glass models. *J. Phys. A Math. Theor.* **15**, 3241 (1982).
8. S. Boixo, T. F. Rønnow, S. V. Isakov, Z. Wang, D. Wecker, D. A. Lidar, J. M. Martinis, M. Troyer, Evidence for quantum annealing with more than one hundred qubits. *Nat. Phys.* **10**, 218–224 (2014).
9. T. F. Rønnow, Z. Wang, J. Job, S. Boixo, S. V. Isakov, D. Wecker, J. M. Martinis, D. A. Lidar, M. Troyer, Defining and detecting quantum speedup. *Science* **345**, 420–424 (2014).
10. D. Venturelli, S. Mandrà, S. Knysh, B. O'Gorman, R. Biswas, V. Smelyanskiy, Quantum optimization of fully connected spin glasses. *Phys. Rev. X* **5**, 031040 (2015).
11. E. Farhi, A. W. Harrow, Quantum supremacy through the quantum approximate optimization algorithm. arXiv:1602.07674 [quant-ph] (2016).
12. S. Hadfield, Z. Wang, B. O'Gorman, E. G. Rieffel, D. Venturelli, R. Biswas, From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms* **12**, 34 (2019).
13. J. S. Otterbach, R. Manenti, N. Alidoust, A. Bestwick, M. Block, B. Bloom, S. Caldwell, N. Didier, E. S. Fried, S. Hong, P. Karalekas, C. B. Osborn, A. Papageorge, E. C. Peterson, G. Prawiroatmodjo, N. Rubin, C. A. Ryan, D. Scarabelli, M. Scheer, E. A. Sete, P. Sivarajah, R. S. Smith, A. Staley, N. Tezak, W. J. Zeng, A. Hudson, B. R. Johnson, M. Reagor, 21 M. P. da Silva, C. Rigetti, Unsupervised machine learning on a hybrid quantum computer. arXiv:1712.05771 [quant-ph] (2017).
14. G. Pagano, A. Bapat, P. Becker, K. S. Collins, A. De, P. W. Hess, H. B. Kaplan, A. Kyprianidis, W. L. Tan, C. Baldwin, L. T. Brady, A. Deshpande, F. Liu, S. Jordan, A. V. Gorshkov, C. Monroe, Quantum approximate optimization of the long-range Ising model with a trapped-ion quantum simulator. *Proc. Natl. Acad. Sci.* **117**, 25396–25401 (2020).
15. M. P. Harrigan, K. J. Sung, M. Neeley, K. J. Satzinger, F. Arute, K. Arya, J. Atalaya, J. C. Bardin, R. Barends, S. Boixo, M. Broughton, B. B. Buckley, D. A. Buell, B. Burkett, N. Bushnell, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, S. Demura, A. Dunsworth, D. Eppens, A. Fowler, B. Foxen, C. Gidney, M. Giustina, R. Graff, S. Habegger, A. Ho, S. Hong, T. Huang, L. B. Ioffe, S. V. Isakov, E. Jeffrey, Z. Jiang, C. Jones, D. Kafri, K. Kechedzhi, J. Kelly, S. Kim, P. V. Klimov, A. N. Korotkov, F. Kostritsa, D. Landhuis, P. Laptev, M. Lindmark, M. Leib, O. Martin, J. M. Martinis, J. R. Mc Clean, M. M. Ewen, A. Megrant, X. Mi, M. Mohseni, W. Mruczkiewicz, J. Mutus, O. Naaman, C. Neill, F. Neukart, M. Y. Niu, T. E. O'Brien, B. O'Gorman, E. Ostby, A. Petukhov, H. Putterman, C. Quintana, P. Roushan, N. C. Rubin, D. Sank, A. Skolik, V. Smelyanskiy, D. Strain, M. Streif, M. Szalay, A. Vainsencher, T. White, Z. J. Yao, P. Yeh, A. Zalcman, L. Zhou, H. Neven, D. Bacon, E. Lucero, E. Farhi, R. Babbush, Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. *Nat. Phys.* **17**, 332–336 (2021).

16. S. Ebadi, A. Keesling, M. Cain, T. T. Wang, H. Levine, D. Bluvstein, G. Semeghini, A. Omran, J. Liu, R. Samajdar, X.-Z. Luo, B. Nash, X. Gao, B. Barak, E. Farhi, S. Sachdev, N. Gemelke, L. Zhou, S. Choi, H. Pichler, S. Wang, M. Greiner, V. Vuletic, M. D. Lukin, Quantum optimization of maximum independent set using rydberg atom arrays. *Science* **376**, 1209–1215 (2022).
17. T. M. Graham, Y. Song, J. Scott, C. Poole, L. Phuttitarn, K. Jooya, P. Eichler, X. Jiang, A. Marra, B. Grinkemeyer, M. Kwon, M. Ebert, J. Cherek, M. T. Lichtman, M. Gillette, J. Gilbert, D. Bowman, T. Ballance, C. Campbell, E. D. Dahl, O. Crawford, N. S. Blunt, B. Rogers, T. Noel, M. Saffman, Multi-qubit entanglement and algorithms on a neutral-atom quantum computer. *Nature* **604**, 457–462 (2022).
18. E. Pelofske, A. Bärtschi, S. Eidenbenz, Quantum Annealing vs. QAOA: 127 Qubit Higher-Order Ising Problems on NISQ Computers. *arXiv:2301.00520* [quant-ph] (2023).
19. S. A. Moses, C. H. Baldwin, M. S. Allman, R. Ancona, L. Ascarrunz, C. Barnes, J. Bartolotta, B. Bjork, P. Blanchard, M. Bohn, J. G. Bohnet, N. C. Brown, N. Q. Burdick, W. C. Burton, S. L. Campbell, J. P. C. III, C. Carron, J. Chambers, J. W. Chan, Y. H. Chen, A. Chernoguzov, E. Chertkov, J. Colina, J. P. Curtis, R. Daniel, M. DeCross, D. Deen, C. Delaney, J. M. Dreiling, C. T. Ertsgaard, J. Esposito, B. Estey, M. Fabrikant, C. Figgatt, C. Foltz, M. Foss-Feig, D. Francois, J. P. Gaebler, T. M. Gatterman, C. N. Gilbreth, J. Giles, E. Glynn, A. Hall, A. M. Hankin, A. Hansen, D. Hayes, B. Higashi, I. M. Hoffman, B. Horning, J. J. Hout, R. Jacobs, J. Johansen, L. Jones, J. Karcz, T. Klein, P. Lauria, P. Lee, D. Liefer, C. Lytle, S. T. Lu, D. Lucchetti, A. Malm, M. Matheny, B. Mathewson, K. Mayer, D. B. Miller, M. Mills, B. Neyenhuis, L. Nugent, S. Olson, J. Parks, G. N. Price, Z. Price, M. Pugh, A. Ransford, A. P. Reed, C. Roman, M. Rowe, C. Ryan-Anderson, S. Sanders, J. Sedlacek, P. Shevchuk, P. Siegfried, T. Skripka, B. Spaun, R. T. Sprenkle, R. P. Stutz, M. Swallows, R. I. Tobey, A. Tran, T. Tran, E. Vogt, C. Volin, J. Walker, A. M. Zolot, and J. M. Pino, A Race Track Trapped-Ion Quantum Processor. *arXiv:2305.03828* [quant-ph] (2023).
20. R. Shaydulin, C. Li, S. Chakrabarti, M. DeCross, D. Herman, N. Kumar, J. Larson, D. Lykov, P. Minssen, Y. Sun, Y. Alexeev, J. M. Dreiling, J. P. Gaebler, T. M. Gatterman, J. A. Gerber, K. Gilmore, D. Gresh, N. Hewitt, C. V. Horst, S. Hu, J. Johansen, M. Matheny, T. Mengle, M. Mills, S. A. Moses, B. Neyenhuis, P. Siegfried, R. Yalovetzky, M. Pistoia, Evidence of Scaling Advantage for the Quantum Approximate Optimization Algorithm on a Classically Intractable Problem. *arXiv:2308.02342* [quant-ph] (2023).
21. S. H. Sack, D. J. Egger, Large-scale quantum approximate optimization on non-planar graphs with machine learning noise mitigation. *arXiv:2307.14427* [quant-ph] (2023).
22. F. B. Maciejewski, S. Hadfield, B. Hall, M. Hodson, M. Dupont, B. Evert, J. Sud, M. S. Alam, Z. Wang, S. Jeffrey, B. Sundar, P. A. Lott, S. Grabbe, E. G. Rieffel, M. J. Reagor, D. Venturelli, Design and execution of quantum circuits using tens of superconducting qubits and thousands of gates for dense ising optimization problems. *arXiv:2308.12423* [quant-ph] (2023).
23. Y. Zhu, Z. Zhang, B. Sundar, A. M. Green, C. H. Alderete, N. H. Nguyen, K. R. A. Hazzard, N. M. Linke, Multi-round QAOA and advanced mixers on a trapped-ion quantum computer. *Quantum Sci. Technol.* **8**, 015007 (2023).
24. E. Farhi, J. Goldstone, S. Gutmann, H. Neven, Quantum Algorithms for Fixed Qubit Architectures. *arXiv:1703.06199* [quant-ph] (2017).
25. Z. Wang, S. Hadfield, Z. Jiang, E. G. Rieffel, Quantum approximate optimization algorithm for Maxcut: A fermionic view. *Phys. Rev. A* **97**, 022304 (2018).
26. K. Marwaha, Local classical MAX-CUT algorithm outperforms $p = 2$ QAOA on high-girth regular graphs. *Quantum* **5**, 437 (2021).
27. J. Wurtz, P. Love, Maxcut quantum approximate optimization algorithm performance guarantees for $p > 1$. *Phys. Rev. A* **103**, 042612 (2021).
28. J. Basso, E. Farhi, K. Marwaha, B. Villalonga, L. Zhou, The Quantum Approximate Optimization Algorithm at High Depth for MaxCut on Large-Girth Regular Graphs and the Sherrington-Kirkpatrick Model. In Le Gall, F. & Morimae, T. (eds.) *17th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2022)*, vol. 232 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 7:1–7:21 (Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2022). URL https://drops.dagstuhl.de/opus/volltexte/2022/16514.
29. E. Farhi, J. Goldstone, S. Gutmann, L. Zhou, The quantum approximate optimization algorithm and the Sherrington-Kirkpatrick model at infinite size. *Quantum* **6**, 759 (2022).
30. K. Marwaha, S. Hadfield, Bounds on approximating Max *k* XOR with quantum and classical local algorithms. *Quantum* **6**, 757 (2022).
31. R. Ayanzadeh, N. Alavisamani, P. Das, M. Qureshi, Frozenqubits: Boosting Fidelity of QAOA by Skipping Hotspot Nodes. *arXiv:2210.17037* [quant-ph] (2022).
32. S. Bravyi, A. Kliesch, R. Koenig, E. Tang, Obstacles to variational quantum optimization from symmetry protection. *Phys. Rev. Lett.* **125**, 260505 (2020).
33. F. Wagner, J. Nüßlein, F Liers, Enhancing Quantum Algorithms for Maximum Cut via Integer Programming. *arXiv:2302.05493* [quant-ph] (2023).
34. R. Ayanzadeh, J. Dorband, M. Halem, T. Finin, Quantum-assisted greedy algorithms. In *IGARSS 2022–2022 IEEE International Geoscience and Remote Sensing Symposium*, 4911–4914 (2022).
35. D. Sherrington, S. Kirkpatrick, Solvable model of a spin-glass. *Phys. Rev. Lett.* **35**, 1792–1796 (1975).
36. A. Montanari, Optimization of the Sherrington-Kirkpatrick Hamiltonian. *arXiv:1812.10897* [quant-ph] (2018).
37. J. Weidenfeller, L. C. Valor, J. Gacon, C. Tornow, L. Bello, S. Woerner, D. J. Egger, Scaling of the quantum approximate optimization algorithm on superconducting qubit based hardware. *Quantum* **6**, 870 (2022).
38. S. Boettcher, Extremal optimization for Sherrington-Kirkpatrick spin glasses. *Eur. Phys. J. B.* **46**, 501–505 (2005).
39. G. Parisi, Infinite number of order parameters for spin-glasses. *Phys. Rev. Lett.* **43**, 1754–1756 (1979).
40. M. J. Schmidt, *Replica Symmetry Breaking at Low Temperatures*. Ph.D. thesis, Julius Maximilians-Universität Würzburg. (2008).
41. M. Aizenman, J. L. Lebowitz, D. Ruelle, Some rigorous results on the Sherrington-Kirkpatrick spin glass model. *Commun. Math. Phys.* **112**, 3–20 (1987).
42. A. Montanari, S. Sen, Semidefinite Programs on Sparse Random Graphs and their Application to Community Detection. *arXiv:1504.05910* [quant-ph] (2015).
43. A. S. W. Afonso, S. Bandeira, D. Kunisky, Computational Hardness of Certifying Bounds on Constrained PCA Problems. *arXiv:1902.07324* [quant-ph] (2019).
44. Z. Cai, R. Babbush, S. C. Benjamin, S. Endo, W. J. Huggins, Y. Li, J. R. McClean, T. E. O'Brien, Quantum Error Mitigation. *arXiv:2210.00921* [quant-ph] (2022).
45. M. Dupont, B. Sundar, Quantum Relax-and-Round Algorithm for Combinatorial Optimization. *arXiv:2307.05821* [quant-ph] (2023).
46. N. Mohseni, P. L. McMahon, T. Byrnes, Ising machines as hardware solvers of combinatorial optimization problems. *Nat. Rev. Phys.* **4**, 363–379 (2022).
47. S. Marsh, J. Wang, A quantum walk-assisted approximate algorithm for bounded NP optimisation problems. *Quantum Inf. Process.* **18**, 61 (2019).
48. A. Bärtschi, S. Eidenbenz, Grover Mixers for QAOA: Shifting Complexity from Mixer Design to State Preparation. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 72–82 (2020).
49. D. Herman, R. Shaydulin, Y. Sun, S. Chakrabarti, S. Hu, P. Minssen, A. Rattew, R. Yalovetzky, M. Pistoi, Portfolio Optimization via Quantum Zeno Dynamics on a Quantum Processor. *arXiv:2209.15024* [quant-ph] (2022).
50. J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth, J. Tennyson, The variational quantum eigensolver: A review of methods and best practices. *Phys. Rep.* **986**, 1–128 (2022).
51. C. J. Morningstar, M. Weinstein, Contractor renormalization group technology and exact Hamiltonian real-space renormalization group transformations. *Phys. Rev. D* **54**, 4131–4151 (1996).
52. D. M. Abrams, N. Didier, B. R. Johnson, M. P. da Silva, C. A. Ryan, Implementation of XY entangling gates with a single calibrated pulse. *Nat. Electron.* **3**, 744–750 (2020).
53. E. Knill, D. Leibfried, R. Reichle, J. Britton, R. B. Blakestad, J. D. Jost, C. Langer, R. Ozeri, S. Seidelin, D. J. Wineland, Randomized benchmarking of quantum gates. *Phys. Rev. A* **77**, 012307 (2008).
54. A. Erhard, J. J. Wallman, L. Postler, M. Meth, R. Stricker, E. A. Martinez, P. Schindler, T. Monz, J. Emerson, R. Blatt, Characterizing large-scale quantum computers via cycle benchmarking. *Nat. Commun.* **10**, 5347 (2019).
55. A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, H. Weinfurter, Elementary gates for quantum computation. *Phys. Rev. A* **52**, 3457–3467 (1995).
56. C. Huang, T. Wang, F. Wu, D. Ding, Q. Ye, L. Kong, F. Zhang, X. Ni, Z. Song, Y. Shi *et al.*, Quantum instruction set design for performance. *arXiv:2105.06074* [quant-ph] (2021).
57. G. Vidal, Efficient simulation of one-dimensional quantum many-body systems. *Phys. Rev. Lett.* **93**, 040502 (2004).
58. M. Dupont, N. Didier, M. J. Hodson, J. E. Moore, M. J. Reagor, Calibrating the classical hardness of the quantum approximate optimization algorithm. *PRX Quantum* **3**, 040339 (2022).
59. M. Dupont, N. Didier, M. J. Hodson, J. E. Moore, M. J. Reagor, Entanglement perspective on the quantum approximate optimization algorithm. *Phys. Rev. A* **106**, 022423 (2022).
60. F. Glover M. Laguna. *Tabu Search*, 2093–2229 (Springer US, Boston, MA, 1998).
61. S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by simulated annealing. *Science* **220**, 671–680 (1983).
62. S. Kirkpatrick, Optimization by simulated annealing: Quantitative studies. *J. Stat. Phys.* **34**, 975–986 (1984).

Acknowledgments

Funding: This work is supported by the Defense Advanced Research Projects Agency (DARPA) under agreement nos. HR00112090058 and IAA 8839, Annex 114. Authors from USRA also acknowledge support under NASA Academic Mission Services under contract no. NNA16BD14C. **Author contributions:** M.D. conceived the project with support from B.E., M.J. H., M.J.R., E.J.R., and D.V. M.D., B.E., M.J.H., S.J., B.S., and Y.Y. developed the code, ran the

experiments, and collected the data. M.D. wrote the manuscript with contributions from B.E., S. H., M.J.R., B.S., and input from all the co-authors. All co-authors contributed to the discussions leading to the completion of this project. The experiments were performed through Rigetti Computing's Quantum Cloud Services QCS using the superconducting quantum processor Rigetti Computing Aspen-M-3 that was developed, fabricated, and operated by Rigetti Computing Inc. **Competing interests:** M.D., B.E., D.F., M.J.H., S.J., M.J.R., and B.S. are, have been, or may in the future be participants in incentive stock plans at Rigetti Computing Inc. M.D. is inventor on two pending patent applications related to this work (no. 63/381,831 and no. 63/487,898). The other authors declare that they have no competing interests. **Data and materials availability:** All experimental data are publicly available at https://doi.org/10.5281/zenodo.7709803.

Submitted 30 March 2023
Accepted 10 October 2023
Published 10 November 2023
10.1126/sciadv.adi0487