

# Evidence of scaling advantage for the quantum approximate optimization algorithm on a classically intractable problem

Ruslan Shaydulin<sup>1\*</sup>, Changhao Li<sup>1</sup>, Shouvanik Chakrabarti<sup>1</sup>, Matthew DeCross<sup>2</sup>, Dylan Herman<sup>1</sup>, Niraj Kumar<sup>1</sup>, Jeffrey Larson<sup>3</sup>, Danylo Lykov<sup>1,4</sup>, Pierre Minssen<sup>1</sup>, Yue Sun<sup>1</sup>, Yuri Alexeev<sup>4</sup>, Joan M. Dreiling<sup>2</sup>, John P. Gaebler<sup>2</sup>, Thomas M. Gatterman<sup>2</sup>, Justin A. Gerber<sup>2</sup>, Kevin Gilmore<sup>2</sup>, Dan Gresh<sup>2</sup>, Nathan Hewitt<sup>2</sup>, Chandler V. Horst<sup>2</sup>, Shaohan Hu<sup>1</sup>, Jacob Johansen<sup>2</sup>, Mitchell Matheny<sup>2</sup>, Tanner Mengle<sup>2</sup>, Michael Mills<sup>2</sup>, Steven A. Moses<sup>2</sup>, Brian Neyenhuis<sup>2</sup>, Peter Siegfried<sup>2</sup>, Romina Yalovetzky<sup>1</sup>, Marco Pistoia<sup>1</sup>

The quantum approximate optimization algorithm (QAOA) is a leading candidate algorithm for solving optimization problems on quantum computers. However, the potential of QAOA to tackle classically intractable problems remains unclear. Here, we perform an extensive numerical investigation of QAOA on the low autocorrelation binary sequences (LABS) problem, which is classically intractable even for moderately sized instances. We perform noiseless simulations with up to 40 qubits and observe that the runtime of QAOA with fixed parameters scales better than branch-and-bound solvers, which are the state-of-the-art exact solvers for LABS. The combination of QAOA with quantum minimum finding gives the best empirical scaling of any algorithm for the LABS problem. We demonstrate experimental progress in executing QAOA for the LABS problem using an algorithm-specific error detection scheme on Quantinuum trapped-ion processors. Our results provide evidence for the utility of QAOA as an algorithmic component that enables quantum speedups.

## INTRODUCTION

Quantum computers have been shown to have the potential to speed up the solution of optimization problems. At the same time, only a small number of algorithmic primitives that provide broadly applicable speedups are known. These include amplitude amplification (1), quantum walks (2–4), and quantum Markov Chain algorithms (5, 6), as well as the recently introduced short path algorithm (7, 8).

The dearth of provable speedups in quantum optimization motivates the development of heuristics. A leading candidate for demonstrating a heuristic speedup in quantum optimization is the quantum approximate optimization algorithm (QAOA) (9, 10). QAOA uses two operators applied in alternation  $p$  times to prepare a quantum state such that, upon measuring it, a high-quality solution to the problem is obtained with high probability. A pair of such operators is commonly referred to as one QAOA “layer.” The state is evolved with a diagonal Hamiltonian encoding the optimization problem by the first operator and with a nondiagonal transverse-field Hamiltonian by the second operator. Here, we consider the evolution times to be hyperparameters that are set by using a fixed, predetermined rule, analogously to the choice of a schedule in simulated annealing.

While QAOA has been studied extensively (11–14), little is known about its potential to provide a scaling advantage over classical solvers. A recent numerical study (13) of random 8-SAT with  $N \leq 20$  variables has shown that the time to solution (TTS) of QAOA with fixed parameters and constant depth grows as  $1.23^N$ . When QAOA is combined with amplitude amplification, the quantum TTS

grows as  $1.11^N$  (13), whereas the best classical heuristic has TTS that grows as  $1.25^N$  (13).

Our work is motivated by this preliminary numerical evidence on small instances, which indicates that QAOA may potentially scale better than classical solvers when executed on an idealized quantum computer.

We study the scaling of QAOA TTS with the problem size on the low autocorrelation binary sequences (LABS) problem (15, 16), also known as the Bernasconi model in statistical physics (17, 18). The LABS problem has applications in communications engineering, where the low autocorrelation sequences are used for designing radar pulses (15, 19). To solve LABS, one has to produce a sequence of  $N$  bits that minimizes a specific quartic objective.

We choose LABS to study the scaling of QAOA TTS for the following three reasons. First, the complexity of LABS grows rapidly, with optimal solutions known only for  $N \leq 66$  and the best heuristics producing approximate solutions of quality decaying with  $N$  for  $N \geq 200$  (20, 21). This makes it a promising candidate problem, since only a few hundred qubits are required to tackle classically intractable instances. Second, the performance of classical solvers for LABS has been benchmarked (20, 21) in terms of the scaling of their TTS with problem size. Since optimal solutions are only known for  $N \leq 66$ , the scaling of TTS for all classical solvers is obtained by fitting results for  $N \leq 66$ . We reproduce these results and observe that the scaling of classical solvers at  $N \leq 40$  matches the behavior for  $N$  up to 66 reported in the literature. This provides evidence that the scaling we observe for QAOA at  $N \leq 40$  will similarly extrapolate to larger  $N$ . Third, LABS has only one instance per problem size  $N$ . Combined with the hardness of LABS, this makes it possible to reliably study the scaling of QAOA at large problem sizes, where simulating tens or hundreds of random instances would be computationally infeasible.

Copyright © 2024 The Authors, some rights reserved; exclusive licensee American Association for the Advancement of Science. No claim to original U.S. Government Works. Distributed under a Creative Commons Attribution License 4.0 (CC BY).

<sup>1</sup>Global Technology Applied Research, JPMorgan Chase, New York, NY 10017, USA.

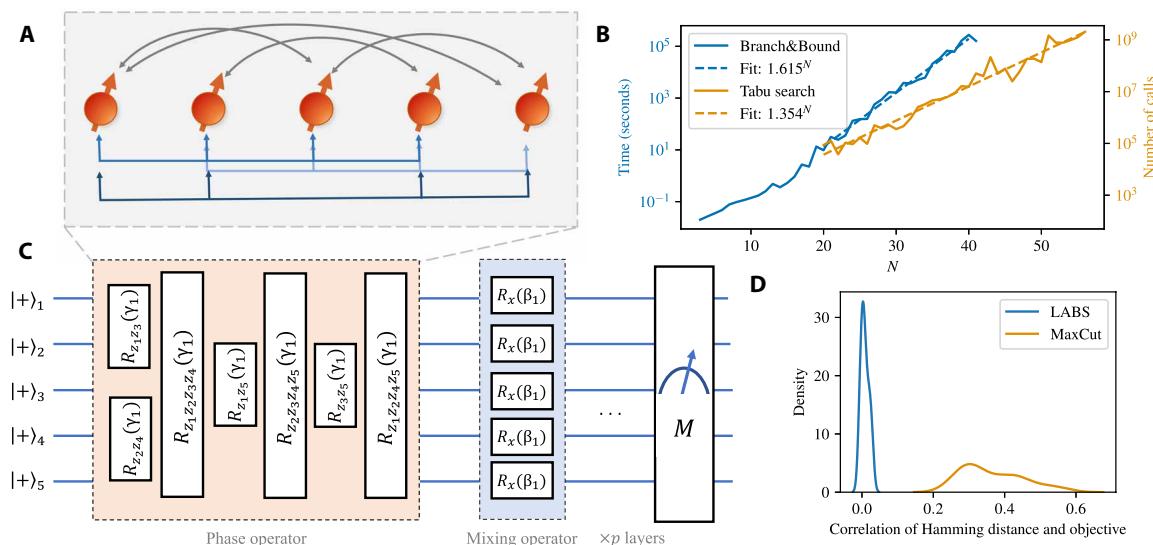
<sup>2</sup>Quantinuum, Broomfield, CO 80021, USA. <sup>3</sup>Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, IL 60439, USA. <sup>4</sup>Computational Science Division, Argonne National Laboratory, Lemont, IL 60439, USA.

\*Corresponding author. Email: ruslan.shaydulin@jpmchase.com

We obtain the scaling by performing noiseless exact simulation of QAOA with fixed schedules. Our results are enabled by a custom algorithm-specific graphics processing unit (GPU) simulator (22), which we execute using up to 1024 GPUs per simulation on the Polaris supercomputer accessed through the Argonne Leadership Computing Facility. We find that the TTS of QAOA with number of layers  $p = 12$  grows as  $1.46^N$ , which is improved to  $1.21^N$  if combined with quantum minimum finding. This scaling is better than that of the best classical heuristic, which has a TTS that grows as  $1.34^N$ . We note that we do not propose any new quantum algorithms in this work. Instead, we study a general quantum optimization heuristic with broad applicability (namely, QAOA) and make no specific modifications to adapt it to the LABS problem.

Our numerical evidence indicates that the proposed quantum algorithm scales better than the best classical heuristic in an idealized setting. However, we do not claim that QAOA is the best theoretically possible algorithm for the LABS problem. In particular, it may be possible to quadratically accelerate the best-known classical heuristic [Memetic Tabu (23)] by applying ideas similar to those used in quantum simulated annealing (5, 24, 25). Nonetheless, our results highlight the potential of QAOA to act as a useful algorithmic component that enables super-Grover quantum speedups.

As a first step toward execution of QAOA for the LABS problem, we implement QAOA on Quantinuum trapped-ion quantum processors (26, 27) on problems with up to  $N = 18$ . We further implement an algorithm-specific error detection scheme inspired by Pauli error detection (28, 29) and demonstrate that it can reduce the impact of noise on solution quality by up to 65%. Our experiments highlight the continuing improvements to quantum computing hardware and the potential of algorithm-specific techniques to reduce the overhead of error detection and correction.



**Fig. 1. Classical and quantum algorithms applied to the LABS problem.** (A) Diagram of the LABS problem (with example of  $N = 5$ ). The problem involves nonlocal two-body (black lines) and four-body (blue lines) interactions. (B) Time to solution (TTS) of classical solvers. For the sizes considered, we observe clear exponential scaling with exponents matching their asymptotic values reported in the literature (see Table 1). (C) Diagram of QAOA circuit for a five-qubit example. Starting from a uniform superposition of the computational basis states, we apply  $p$  layers of phase and mixing operators, followed by measurement in the computational basis. (D) Distribution over  $21 \leq N \leq 31$  (for LABS) and 34 random instances (for MaxCut on random three-regular graphs with 20 nodes) of Pearson product-moment correlation coefficients relating the Hamming distance of bitstrings from the optimal solution with the objective value of the bitstring. LABS has a much lower correlation between the Hamming distance and objective, indicating that it is much harder than the commonly considered MaxCut problem.

## RESULTS

### Problem statement

We begin by formally defining the LABS problem, discussing the state of the art of classical LABS solvers, and describing how QAOA is applied to solve the problem.

For a given sequence of spins  $s_i \in \{\pm 1\}$ , the autocorrelation is given as

$$\mathcal{A}_k(\mathbf{s}) = \sum_{i=1}^{N-k} s_i s_{i+k} \quad (1)$$

The goal of the LABS problem is to find a sequence of spins that minimizes the so-called “sidelobe” energy

$$\mathcal{E}_{\text{sidelobe}}(\mathbf{s}) = \sum_{k=1}^{N-1} \mathcal{A}_k^2(\mathbf{s}) \quad (2)$$

or, equivalently, maximizes the merit factor

$$\mathcal{F}(\mathbf{s}) = \frac{N^2}{2\mathcal{E}_{\text{sidelobe}}(\mathbf{s})} \quad (3)$$

The TTS is defined as the time a solver takes to produce this sequence. The energy  $\mathcal{E}_{\text{sidelobe}}(\mathbf{s})$  is a polynomial containing terms of degrees 2 and 4, visualized in Fig. 1A. An instance of the LABS problem is unique for each  $N$  and can be encoded on qubits by the following Hamiltonian

$$H_C = 2 \sum_{i=1}^{N-3} z_i \sum_{t=1}^{\lfloor \frac{N-i-1}{2} \rfloor} \sum_{k=t+1}^{N-i-t} z_{i+t} z_{i+k} z_{i+k+t} + \sum_{i=1}^{N-2} z_i \sum_{k=1}^{\lfloor \frac{N-i}{2} \rfloor} z_{i+2k} \quad (4)$$

where  $z_j$  is a Pauli  $z$  operator acting on qubit  $j$ .

The runtimes of state-of-the-art classical solvers for the LABS problem scale exponentially, with clear exponential scaling present at  $N \leq 40$  as shown in Fig. 1B. The best-known exact solvers are branch-and-bound methods that have a running time that scales as  $1.73^N$  (21). The best-known heuristic for general LABS is tabu search initialized with a memetic algorithm (Memetic Tabu) (23), and has a running time that scales as  $1.34^N$  (30). We provide a survey of classical solvers for the LABS problem in the Supplementary Materials.

To see why LABS is harder to solve than other commonly studied problems such as MaxCut, we can examine the correlation between the Hamming distance to the optimal solution and the objective. The comparison is shown in Fig. 1C. This correlation is one example of problem structure used by both classical and quantum heuristics to solve the problem quickly (9). The absence of this correlation highlights the hardness of LABS compared with other commonly considered problems such as MaxCut.

As a consequence of the exponential scaling, the LABS problem becomes classically intractable at moderate sizes. Specifically, the value of the best-known merit factor decreases notably for high  $N$ , whereas the asymptotic limit predicts that the merit factor should stay approximately constant. This failure of state-of-the-art heuristics has been observed for  $N > 200$  (20, 21). The clear failure of the classical method to obtain high-quality solutions even at small sizes makes LABS an appealing candidate problem for quantum optimization heuristics.

Here, we tackle the LABS problem using QAOA. As shown in the circuit diagram Fig. 1D, QAOA solves optimization problems by preparing a parameterized state

$$|\beta, \gamma\rangle = \prod_{l=1}^p e^{-i\beta_l} \sum_{j=1}^N x_j e^{-i\gamma_l H_C} |+\rangle^{\otimes N} \quad (5)$$

where  $|+\rangle^{\otimes N}$  is a uniform superposition over computational basis states,  $H_C$  is the diagonal Hamiltonian encoding the problem, and  $x_j$  is a Pauli  $x$  operator acting on qubit  $j$ . The operator  $e^{-i\gamma_l H_C}$  is commonly referred to as the phase operator, and  $e^{-i\beta_l \sum_{j=1}^N x_j}$  as the mixing operator. The evolution times  $\beta, \gamma$  are hyperparameters chosen to maximize some figure of merit, such as the expected quality of the measurement outcomes or the probability of measuring the optimal solution. While  $\beta, \gamma$  can be optimized independently for each problem size, we consider them to be hyperparameters and use one fixed set of parameters for the LABS problem with a given QAOA depth  $p$  regardless of size. The fixed set of parameters is obtained by optimizing  $\beta, \gamma$  numerically for a number of small problem sizes and introducing

an averaging and rescaling procedure to extrapolate parameters to any problem size (see Materials and Methods).

When choosing the parameters  $\beta, \gamma$  and evaluating the quality of the solution obtained by QAOA, two figures of merit are commonly considered. The first one is the expected merit factor of the sampled binary strings, given by

$$\langle C \rangle_{MF} = \langle \beta, \gamma | \frac{N^2}{2H_C} | \beta, \gamma \rangle = \sum_{s \in \{0,1\}^N} \text{Pr}(s) \mathcal{F}(s) \quad (6)$$

We will refer to  $\langle C \rangle_{MF}$  as the “QAOA energy” as a shorthand. The second figure of merit is the probability of sampling the exact optimal solution, denoted by  $p^{\text{opt}}$  and equal to the sum of squared absolute values of amplitudes of basis states corresponding to exactly optimal solutions.

In the numerical experiments below, we follow the protocol of (13) and focus on scaling of the QAOA TTS with problem size  $N$  as the QAOA depth  $p$  is held constant. QAOA TTS is defined as  $\frac{1}{p^{\text{opt}}}$ , i.e., the expected number of measurements required to obtain an optimal solution from the QAOA state. Boulebnane and Montanaro (13) rigorously show that, for random  $k$ -SAT, the runtime of constant-depth QAOA grows exponentially with  $N$  at any fixed  $p$ , with the scaling exponent depending on  $p$ . While the nature of the LABS problem makes it difficult to obtain analytical results analogous to (13), our numerical results also show clear exponential scaling of TTS. We note that, in practice, TTS of QAOA is  $\Theta(N^2)^{\frac{1}{p}}$ , where the  $\Theta(N^2)$  prefactor comes from the cost of implementing the LABS phase oracle (31). However, we do not include it in our analysis because it does not affect the scaling exponent.

### Scaling of quantum TTS for LABS problem

We now present the numerical results demonstrating the scaling of TTS of QAOA and QAOA augmented with quantum minimum finding (“QAOA+QMF”). The results are summarized in Table 1. Throughout this section, we present the numerical results obtained using exact noiseless simulations. The runtime scaling is obtained by evaluating QAOA once with fixed parameters  $\beta, \gamma$  (i.e., with no overhead of parameter optimization) and computing the value  $p^{\text{opt}}$  with high precision. We discuss the parameter setting procedure and the details of simulation in Materials and Methods.

We are interested in the scaling of the runtime of QAOA for large problem sizes  $N$ . An important question to address is the choice of the smallest  $N$  to include in the scaling analysis, since the algorithm’s behavior at small sizes may not be predictive of its behavior at large

**Table 1. Scaling exponents for quantum and classical algorithms.** Confidence intervals (CIs) are 95%. The reported asymptotic exponential scaling of classical state-of-the-art solvers are reproduced at  $N \leq 40$ . For branch-and-bound, we include both the time to obtain a certificate of optimality [TTO reported in (21)] and the much shorter time to find an optimal solution (TTS). We observe that QAOA with constant depth of  $p = 12$  augmented with quantum minimum finding (“QAOA+QMF”) has better TTS scaling than the best-known classical heuristics. Best scaling is indicated in boldface.

	QAOA+QMF	QAOA	Memetic Tabu		Branch-and-bound		TTO (21)
			Reproduced	(23, 30)	Reproduced	TTS	
Fit	<b>1.21</b>	1.46	1.35	1.34	1.62	1.76	1.73
CI	(1.19, 1.23)	(1.42, 1.50)	(1.33, 1.38)	N/A	(1.57, 1.66)	(1.72, 1.79)	N/A

sizes. Note that the largest  $N$  we include is limited by the capability of the classical simulator. We use the quality of the fit as the criterion for the choice of the cutoff on  $N$ . Figure 2A shows that if we set the cutoff at  $N \geq 28$ , we obtain a robust high-quality fit ( $R^2 > 0.94$ ), with the quality of the fit remaining stable as  $p$  grows. On the other hand, if smaller  $N$  values are included, the quality of fit begins to decay with  $p$ . Therefore, we include only  $N \geq 28$ , obtaining the fit presented in Fig. 2B. We observe that TTS of QAOA grows as  $1.46^N$  with problem size at constant QAOA depth  $p = 12$ . We present evidence that the scaling exponent for QAOA at  $p = 12$  is not sensitive to the choice of  $N_{\min}$  in the Supplementary Materials.

As a quantum optimization heuristic with constant depth, on a fault-tolerant quantum computer, the QAOA performance can be improved by using amplitude amplification (13, 31) or, more specifically, quantum minimum finding (32) (see Materials and Methods). The resulting scaling of TTS of QAOA augmented with quantum minimum finding (“QAOA+QMF”) is  $1.21^N$ .

We observe that, beyond a certain value ( $p \approx 12$ ), increasing QAOA depth does not lead to better scaling of TTS. This behavior is demonstrated in Fig. 2C. Consequently, running QAOA with  $p$  higher than 12 does not give any scaling advantage over amplitude amplification. This behavior is illustrated in Fig. 3A, which shows the increase in the success probability  $p^{\text{opt}}$  from applying a given step of QAOA and amplitude amplification. For amplitude amplification, at step  $p$ , we have  $p^{\text{opt}} = \{\sin[(2p+1)\arcsin\sqrt{p_0}]\}^2$ , where  $p_0 = \frac{8}{2^N}$  is the initial (random guess) success probability (33). Note that the 8 in the numerator is a consequence of a dihedral group symmetry, namely,  $D_4$ . While asymptotically equivalent, amplitude amplification performs better than a realistic generalized minimum finding algorithm (32), as the formula used here considers the scenario where we know which states to amplify (i.e., the optimal merit factor is known). We observe that for small  $p$ , a step (layer) of QAOA gives orders of magnitude larger increase in success probability than does a step of amplitude amplification, implying an even larger improvement over direct application of quantum minimum finding. We provide additional details on comparison between QAOA and amplitude amplification in the Supplementary Materials.

We observe that the QAOA dynamics with parameters optimized for expected solution quality  $\langle C \rangle_{\text{MF}}$  and success probability  $p^{\text{opt}}$  are different. We plot the optimized parameters in Fig. 3B. We note that the parameters optimized with respect to one metric give performance that is far from optimal with respect to the other metric. This can be seen in Fig. 3C, which plots the energy distribution (with

respect to the cost Hamiltonian) of the states appearing in the QAOA wave function weighted by probability. With the parameters optimized for  $|C\rangle_{\text{MF}}$ , the QAOA output distribution is concentrated around its mean, and the overlap with the ground state or  $p^{\text{opt}}$  is very small. On the other hand, when the parameters are optimized with respect to  $p^{\text{opt}}$ , the wave function is not concentrated and has large probability weight on the target ground state (i.e., high  $p^{\text{opt}}$ ). This comes at the cost of substantial overlap with high-energy states, which leads to poor expected solution quality. In the Supplementary Materials, we discuss the behavior of QAOA with parameters optimized with respect to different objectives.

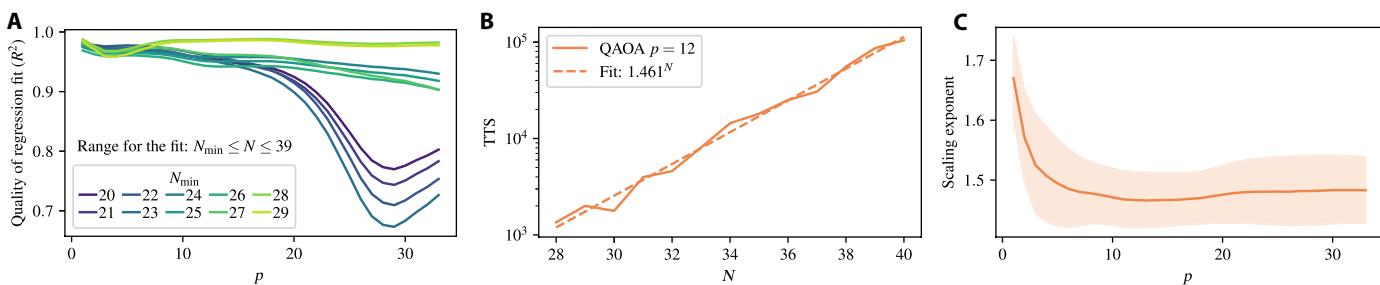
## Experiments on trapped-ion system

We now present the experimental results demonstrating the algorithmic and hardware progress toward the practical implementation of QAOA. Implementation of the phase operator is especially challenging for currently available quantum processors. It requires a large number of geometrically nonlocal two-qubit gates, demanding high gate fidelity.

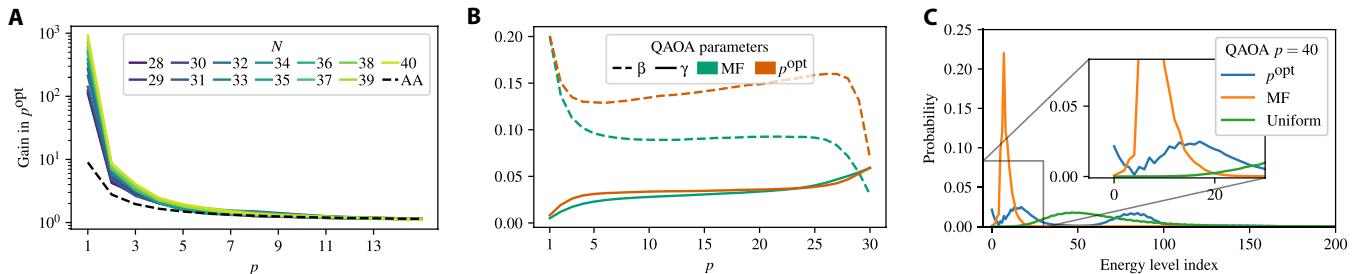
Recent progress in trapped-ion platforms based on the quantum charge-coupled device (QCCD) architecture (26, 27) has led to a rapid increase in the number of qubits while maintaining high fidelity, enabling large-scale QAOA demonstrations (34, 35). These systems implement two-qubit gates between arbitrary pairs of qubits by transporting ions into physically separate gate zones, resulting in high-fidelity two-qubit gates with low cross-talk. We leverage this progress to execute QAOA circuits for the LABS problem on Quantinuum H-series trapped-ion systems.

To implement the QAOA circuit shown in Fig. 1D, we have to implement the phase operator. The four-body terms in the phase operator are decomposed into *cnot* gates and the native  $R_{zz}(\theta) = e^{-i\frac{\theta}{2}zz}$  rotation as shown in Fig. 4A. To reduce the cost of implementing both the two-qubit and four-qubit interaction terms, we optimize the circuit by greedily canceling *cnot* gates (for algorithm details and gate count reduction, see the Supplementary Materials). The resulting circuit containing *cnots* and  $R_{zz}$ s is then transpiled into the two-qubit  $R_{zz}$  gates and single-qubit gates that can be natively implemented by the trapped-ion system. We remark that the number of two-qubit gates is  $\approx 10^3$  at  $N = 18$ , putting our experiments among the largest quantum optimization demonstrations on quantum hardware to date (36–38).

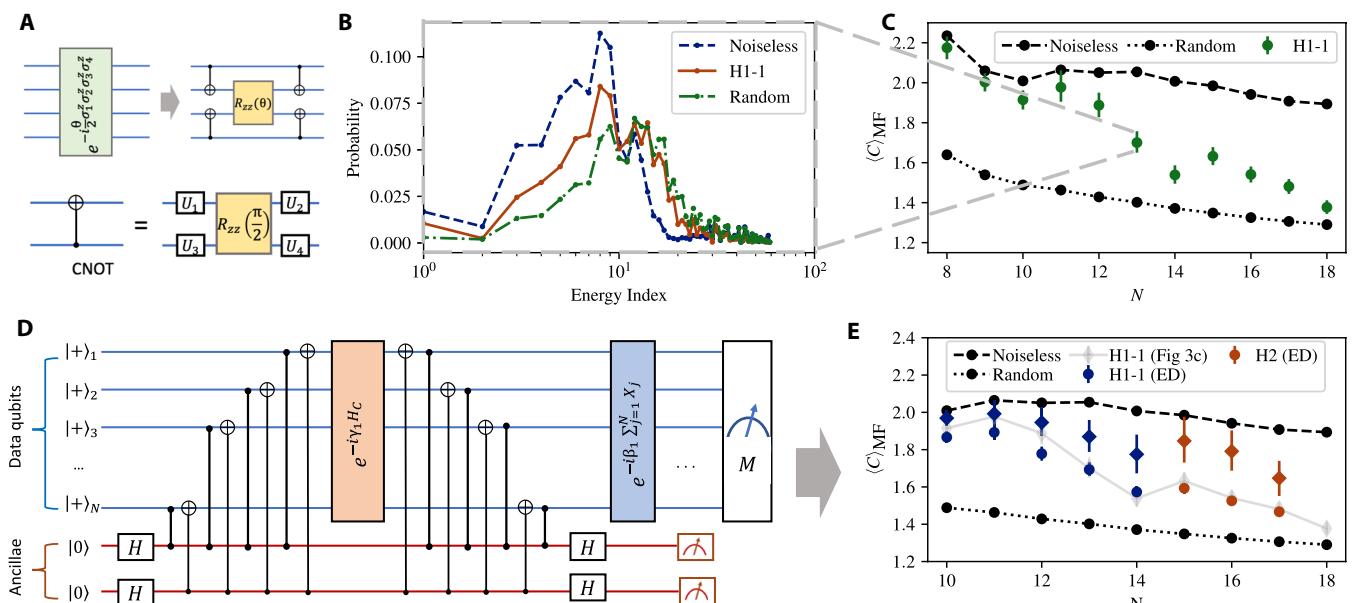
Here, we execute QAOA circuits with  $p = 1$  using parameters  $\beta, \gamma$  optimized in noiseless simulation, followed by a projective



**Fig. 2. QAOA runtime scaling.** (A) Quality of the exponential fit for different choices of minimum  $N$  to include in the fit.  $N \geq 28$  results in a robust fit, the quality of which does not deteriorate with  $p$ .  $N = 40$  is omitted as it was only simulated up to  $p = 22$ . (B) TTS of QAOA at  $p = 12$ . Clear exponential scaling is observed. (C) Scaling exponent of QAOA runtime for different QAOA depths  $p$ . Shaded area shows 95% confidence interval. Increasing  $p$  beyond  $p \approx 12$  does not lead to better scaling.



**Fig. 3. QAOA dynamics under different parameter schedules.** (A) Gain in success probability  $p^{\text{opt}}$  from applying step  $p$  of QAOA and amplitude amplification (AA). The gain is defined as  $p_{\text{at step } p}^{\text{opt}} / p_{\text{at step } (p-1)}^{\text{opt}}$ . The gain at  $p = 1$  is over the random guess. Only one line is plotted for amplitude amplification since the lines for the values of  $N$  considered are visually indistinguishable. For small  $p$ , a QAOA layer gives orders of magnitude larger gain than a step of AA. (B) Fixed QAOA parameters for  $p = 30$  chosen with respect to the QAOA energy ( $\langle C \rangle_{\text{MF}}$ ) ("MF") and probability of sampling the optimal solution (" $p^{\text{opt}}$ "). Different choice of optimization objective gives different resulting parameters. (C) Probability of obtaining a binary string corresponding to a given energy level of the LABS problem (the zeroth energy level is the ground state or optimal solution; lower is better). When parameters are optimized with respect to the expected merit factor (labeled "MF"), the QAOA output state is concentrated around the mean and fails to obtain a high overlap with the ground state. On the other hand, when parameters are optimized with respect to  $p^{\text{opt}}$  (labeled " $p^{\text{opt}}$ "), the QAOA state has a high overlap with both the ground state and higher energy states. The probability of obtaining the ground state is 27.3 times greater for QAOA with parameters optimized with respect to  $p^{\text{opt}}$  at  $p = 40$ .



**Fig. 4. Experimental results on trapped-ion system.** (A) Decomposition of four-body interaction terms into a two-body  $R_{zz}$  gate and four two-body  $cnot$  gates, which can be realized via native  $R_{zz}$  gates. (B) Energy density plot from experimental measured bitstrings for  $N = 13$ . Energy index is arranged in energy ascending order. As a comparison, the distributions for noiseless  $p = 1$  QAOA simulation and random guess (assuming uniform distribution of all possible bitstrings) are shown. (C) Experimental results up to 18 qubits on a trapped-ion quantum device (H1-1) with QAOA layer  $p = 1$  with optimized QAOA parameters. The error bars are calculated with 99% confidence intervals hereafter. (D) Illustration of parity check circuit. The z and x parities of states are mapped to ancillary qubits after implementation of full (or part of) phase operators via  $cz$  and  $cnot$  gates, respectively, followed by mid-circuit measurement on the ancillary qubits to extract the parity syndrome result. (E) Experimental results for circuit with parity check. Three mid-circuit z-parity and x-parity checks were performed using six ancillary qubits. The ancillae can also be reused after appropriate reset during the circuit. The red data points were run on the Quantinuum H2 hardware, while the blue data were from the H1-1 device. Data run on the H1-1 device without any ancillary qubits are shown in gray. Circles (diamonds) are the data without (with) postselection. The abbreviation ED refers to the error detection via the parity checks. Number of mid-circuit parity checks is fixed to be two for  $N = 10, 11$  and three for all other  $N$ . Improvement in expected merit factor after postselection according to parity syndrome measurement is observed.

measurement in the computational basis. In Fig. 4B, we show the energy probability distribution of measured bitstrings for  $N = 13$ . We observe a broad distribution due to the limited number of layers and experimental imperfections. Nevertheless, even at high  $N$ , where two-qubit gate count is high and the gate errors can be substantial, we observe a clear signal that indicates that QAOA is

outperforming random guess. This is shown in Fig. 4C, which presents the experimentally obtained expected merit factors for various problem size up to  $N = 18$ . We note that the merit factor drops quickly for larger  $N$  and is approaching random guess because of experimental imperfections. We also note that at this scale LABS is easy for classical heuristics, which obtain optimal

merit factors in  $<1$  s. Implementing QAOA for LABS instances that are hard for classical solvers would likely require error correction, as the current implementation leads to an estimated two-qubit gate count of  $\approx 7.5 \times 10^5$  already at  $N = 67$  and  $p = 12$ .

To improve the performance in the presence of noise, we implement an algorithm-specific error detection scheme. Since only the phase operator requires two-qubit gates, we focus on detecting errors that occur in the corresponding part of the circuit. Our scheme is based on the Pauli sandwiching error-detecting procedure of (28), which uses pairs of parity checks to detect some but not necessarily all errors that occur in a given part of the circuit. Following (39, 40), we use the symmetries of the optimization problem to construct the parity checks. Specifically, we note that the LABS Hamiltonian preserves both  $z$  and  $x$  parities, that is,  $[H_C, \bigotimes_i^N z_i] = [H_C, \bigotimes_i^N x_i] = 0$ . We compute the parities onto ancillary qubits and perform mid-circuit measurement to determine whether an odd number of  $z$ - or  $x$ -flip errors occur during the circuit execution. The circuit with one check is shown in Fig. 4D. In the hardware experiments shown in Fig. 4E, we use up to three parity checks and observe consistent improvements in QAOA performance after postselecting on their outcomes. After postselection, the difference of merit factor between experimental results and noiseless simulation is reduced by 54% on average and up to 65% for specific  $N$ . In the Supplementary Materials, we present additional details on the error-detecting scheme performance, including how performance improves with the number of parity checks and the reduction in the algorithm runtime. We note that while error detection does not directly give samples with better merit factors, the potential improvement in runtime can be translated into performance gains at the algorithm level, for example, by being able to take more samples within a given time budget. In our experiments, in all but two cases, the optimal bitstring could be found within the postselected sample and, in all cases, within the total sample.

## DISCUSSION

Our main finding is that quantum minimum finding enhanced with QAOA scales better than the best-known classical heuristics for the LABS problem. This provides evidence for the potential of QAOA to act as a building block that provides algorithmic speedups on an idealized fault-tolerant quantum computer. We envision QAOA being used in a variety of algorithmic settings, similarly to how amplitude amplification acts as a subroutine in quantum algorithms for backtracking, branch-and-bound, and so on.

While our numerical evidence is only obtained by fitting instances with  $N \leq 40$ , there are three observations that make us optimistic the observed scaling will hold for larger  $N$ . First, evaluating TTS is only possible when the solution is known, i.e., only up to 66 variables. This limitation applies equally to the classical solvers, which also report their scaling on  $N \leq 66$ . Therefore, for the purposes of honest comparison between classical and quantum solvers, the relevant problem size is up to 66 variables, out of which our data cover up to 40 variables. Second, we note that for classical solvers considered, there is no change in scaling between  $N \leq 40$  and  $40 \leq N \leq 66$  (see Fig. 1B). This supports our claim that QAOA scaling will also remain the same up to at least  $N = 66$ . Finally, we note that there is a rich literature showing that QAOA performance at small finite  $N$  matches the rigorously derived infinite-size-limit behavior. This includes results for MaxCut (41), Sherrington-Kirkpatrick

model (12),  $k$ -spin models (42), and random  $k$ -SAT (13). The results for  $k$ -SAT specifically focus on the scaling of TTS (13), matching our setting.

We take the first step toward the execution of QAOA for the LABS problem by implementing an algorithm-specific error detection scheme on a trapped-ion quantum processor. However, further improvements in quantum error correction and hardware are necessary to implement the quantum minimum finding augmented with QAOA. In particular, the overheads of fault tolerance (43) must be substantially reduced to realize the quantum speedup.

## MATERIALS AND METHODS

### Quantum minimum finding enhanced with QAOA

Here, we present the scaling results for QAOA combined with amplitude amplification (AA) or, more specifically, with quantum minimum finding (“QAOA+QMF” in Table 1). This reduces the scaling exponent by half as compared to directly sampling QAOA output. We now discuss in detail how QAOA is combined with the generalized quantum minimum finding algorithm of (32) to obtain the stated scaling.

We begin by noting that standard AA is not sufficient. This is because the LABS problem is framed as optimization and not search, i.e., there is no oracle for marking a global minimum. The trick for handling optimization is to perform a standard reduction from optimization to feasibility. The reduction is performed by introducing a threshold on the cost as a constraint and performing a binary search using AA as a subroutine. The oracle used by AA marks the elements below the current threshold. This reduction was first introduced by Dürr and Hoyer (DH) (1). However, the quantum minimum finding algorithm of DH utilizes standard Grover search, i.e., it requires the initial state to be the uniform superposition. A modification to it is required to leverage the improved success probability afforded by QAOA.

Van Apeldoorn *et al.* (32) provided a simple extension of DH that allows arbitrary initial states, with the overall cost scaling inversely with overlap between the initial state and state encoding the optimal solution. We leverage this extension in our quantum algorithm. We use constant-depth QAOA to prepare the initial state for the quantum minimum finding algorithm. As QAOA state has overlap with the optimal state that is much larger than that of uniform superposition and scales more favorably, we obtain better performance than the direct minimum finding of DH. Specifically, we provide numerical evidence that our algorithm obtains a super-Grover speedup over exhaustive search for the LABS problem and scales better than the best-known classical heuristics. We present our modification to include QAOA for outputting an optimal solution  $x^*$  to the LABS problem in Algorithm 1. It is based on the generalized minimum finding procedure outlined in lemma 48 of (32). To keep the current work self-contained, we include the analysis of the algorithm below. We will use the following standard quantum subroutine based on Grover search that searches for an element with unknown probability in a quantum state.

**Lemma 1.** [Exponential quantum search, (44)]. Let  $|\psi\rangle = U|0\rangle^{\otimes N}$  be a quantum state in a  $2^N$ -dimensional Hilbert space with computational basis elements indexed by  $N$ -bit bitstrings, and  $m : \{0,1\}^N \rightarrow \{0,1\}$  be a marking function such that  $\sum_{\{x|m(x)=1\}} |\langle \psi|x \rangle|^2 \geq p$ . There

exists a quantum algorithm **EQSearch**( $U, m, \delta$ ) that outputs an element  $x^*$  such that  $m(x^*) = 1$  with probability at least  $\delta$  using  $O\left[\frac{1}{\sqrt{p}} \log\left(\frac{1}{\delta}\right)\right]$  applications of  $U$  and  $m$ .

### Algorithm 1 QAOA Enhanced with Quantum Minimum-Finding

```

Require: Unitary  $U_{\text{QAOA}}$  acting on  $\mathbb{C}^{2^N}$  such that  $|\langle x^* | U_{\text{QAOA}} | 0 \rangle^{\otimes N}| \geq 1/\sqrt{p_{\text{opt}}}$  for unknown  $p_{\text{opt}}$ ,  $V_{\text{LABS}}$  for computing  $\mathcal{E}_{\text{sidelobe}}$  into a register, and  $\delta \in (0, 1)$ , positive number  $M \leq 2^N$ ,  $C$  is the constant corresponding to the  $O(\cdot)$  in Lemma 1
Ensure: If  $M$  is greater than  $1/\sqrt{p_{\text{opt}}}$ , output  $x^*$  with  $\geq 1 - \delta$  probability using  $O(\log(1/\delta)M)$  calls to  $U_{\text{QAOA}}$  and  $V_{\text{LABS}}$  (and their inverses).
 $x_{\text{res}}$  is set to an empty list.
for  $i \leftarrow 1$  to  $\lceil \log(1/\delta) \rceil$  do
     $t \leftarrow 0$ ;  $s_0 \leftarrow \infty$ 
    while number of calls to  $U_{\text{QAOA}}$  &  $V_{\text{LABS}}$  is  $< 3CMN$  do
         $t \leftarrow t + 1$ 
        Define  $m_t : \{0, 1\}^N \rightarrow \{0, 1\}$  such that  $m_t(x) = 1$  if and only if  $\mathcal{E}_{\text{sidelobe}}(x) < s_{t-1}$ . Note that  $m_t$  can be coherently evaluated using one query to  $V_{\text{LABS}}$ .
        Set  $s_t = \text{EQSearch}(U_{\text{QAOA}}, m_t, 1/(6 \cdot 2^N))$ .
    end while
    Append  $s_t$  to  $x_{\text{res}}$ .
end for
Output minimum of  $x_{\text{res}}$ .
```

**Theorem 1.** Suppose a constant-depth QAOA circuit  $U_{\text{QAOA}}$  prepares a state  $|\psi\rangle = U_{\text{QAOA}}|0\rangle^{\otimes N}$  with  $N \geq 3$  such that we have  $|\langle x^* | \psi \rangle| \geq 1/\sqrt{p_{\text{opt}}}$ , where  $|x^*\rangle$  encodes an optimal solution to the  $N$ -bit LABS problem in a computational basis state, and we assume that  $p_{\text{opt}} \geq 1/N$ . Then, running Algorithm 1 with parameters  $M \geq 1/\sqrt{p_{\text{opt}}}$  and failure probability  $\delta$  runs with a gate complexity of  $O(\text{poly}(N) \log(1/\delta)M)$  and finds  $x^*$  with probability at least  $1 - \delta$ .

*Proof.* See the Supplementary Materials.

### Choice of QAOA parameters $\beta, \gamma$

Our strategy for setting the QAOA parameters  $\beta, \gamma$  used in our experiments is twofold. First, we optimize QAOA parameters for small  $N$  using the FOURIER reparameterization scheme of (11). Second, we use the optimized parameters for small  $N$  to compute fixed

QAOA parameters that are then used for larger  $N$ . To apply the fixed parameters to an instance with a given size  $N$ , we rescale the parameters  $\gamma$  by  $N$ . We discuss the parameter optimization scheme and the parameter rescaling in the Supplementary Materials. We note that the results presented above can be improved if better parameter setting strategies are used.

The procedure for obtaining the set of fixed QAOA parameters is visualized in Fig. 5. Specifically, we optimize QAOA parameters for a set of small instances with sizes  $\{N_j\}_{j=1}^M$  attainable in simulation and set the fixed parameters to be the mean over the optimized parameters

$$\beta^{\text{Fixed}} = \frac{1}{M} \sum_{j=1}^M \beta_{N_j}^* \quad (7)$$

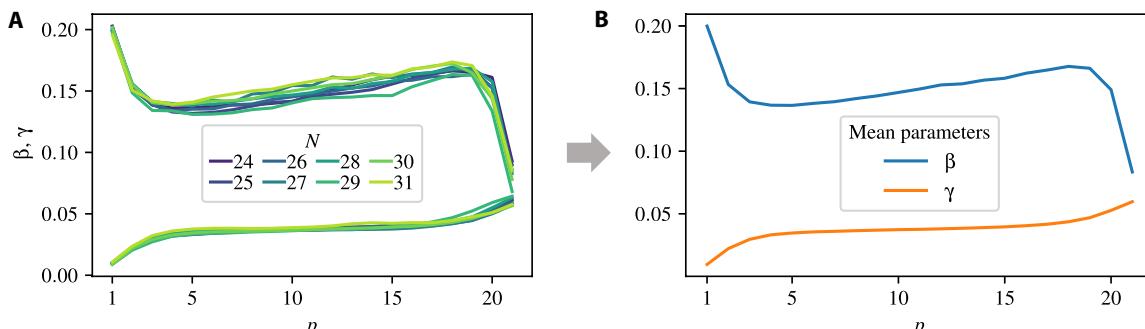
$$\gamma^{\text{Fixed}} = \frac{1}{M} \sum_{j=1}^M N_j \gamma_{N_j}^* \quad (8)$$

where  $\beta_{N_j}^*, \gamma_{N_j}^*$  are the QAOA parameters optimized for the LABS instance of size  $N_j$  and  $M$  is the number of optimized instances. Then, the parameters used in QAOA for size  $N$  are given by  $\beta^{\text{textFixed}}, \gamma^{\text{Fixed}} / N$ . We use  $24 \leq N_j \leq 31$  ( $M = 8$ ).

### Error detection by symmetry verification

The error detection scheme relies on the symmetry of phase operator defined by Eq. 2. As it commutes with both  $\otimes_i^N z_i$  and  $\otimes_i^N x_i$  operators, one can measure the value of these operators and perform postselection on the measurement outcomes. That is, the state after the phase operator should have the same  $z$  and  $x$  parity as before it. In the presence of an odd number of bit flip or phase flip errors that occur during the implementation of phase operators, the resulting state will not be in the  $+1$  eigenspace of the two syndrome operators.

Experimentally, we divide the whole phase operator into  $m$  splits such that each split has approximately the same number of two-qubit gates, and we perform syndrome checks at the end of each split to detect errors. The syndrome operators are mapped to ancillary qubits via sequential controlled- $x$  or controlled- $z$  gates and Hadamard gates applied before and after the partial phase operator. Since the number of two-qubit gates for the phase operators is higher than the number of gates used for the mapping by two to three orders of magnitude, additional errors introduced by ancillae are negligible.



**Fig. 5. Visualization of how the fixed parameters are obtained.** (A) Optimized QAOA parameters  $\beta$  (top lines) and  $\gamma$  (bottom lines) for  $p = 21$ .  $\gamma$  is multiplied by  $N/24$  (constant factor of  $\frac{1}{4}$  added for figure readability in both subfigures). (B) Fixed parameters obtained by taking the arithmetic mean over the optimized parameters.

Furthermore, the cross-talk error probability during mid-circuit measurements is on the order of  $10^{-5}$ , considerably lower than the typical two-qubit gate infidelity of  $2 \times 10^{-3}$  for the trapped-ion systems we used (27). As a result, our error detection scheme leads to large improvements in QAOA performance on hardware at the cost of the number of repetitions growing exponentially with the number of checks (28). We note that the performance of the error detection scheme can be further improved by implementing parity checks using fault-tolerant constructions (45).

### Scaling of classical solvers

All scaling coefficients are obtained by fitting a least-squares linear regression on the logarithm of TTS. The confidence intervals on the scaling coefficients are obtained by using the Student's *t* distribution and are reported with 95% confidence.

We use commercial state-of-the-art branch-and-bound solvers in numerical experiments. Figure 1B and Table 1 show results obtained using Gurobi (46), although we obtain similar results for CPLEX (47) (see the Supplementary Materials). The use of commercial branch-and-bound solvers is motivated by the observation that their scaling closely matches that reported in (21). Specifically, we observe that for both solvers the time to produce a certificate of optimality (TTO) scales with an exponent within a 95% confidence interval of the 1.73 exponent reported in (21). We note that unlike the solver presented in (21), commercial solvers are not parallelizable and can take advantage of only one central processing unit (CPU) with at most tens of cores. Since QAOA is a heuristic and does not guarantee optimality, we additionally run branch-and-bound solvers until a solution with an exactly optimal merit factor is found, at which point the execution is stopped. The resulting TTS scales more favorably: For Gurobi, the scaling is  $1.615^N$ , with a 95% confidence interval of (1.571, 1.659). All the numbers reported correspond to the mean CPU time, with the mean taken over 100 random seeds for  $N \leq 32$  and 10 random seeds for  $N > 32$ . We present additional details of classical solver benchmarking in the Supplementary Materials.

Branch-and-bound algorithms are the best-known exact solvers for the LABS problem. In the regime where proving optimality is out of reach and the goal is simply to efficiently obtain sequences with high merit factors, heuristic algorithms are preferable. The best runtimes and runtime scaling reported in the literature (23) are from an algorithm known as Memetic Tabu. Memetic Tabu is a memetic algorithm, that is, an evolutionary algorithm augmented by local search. Specifically, an evolutionary algorithm is used to find initializations for tabu search, a metaheuristic that augments local neighborhood search with a data structure (known as the tabu list) that filters possible local moves if the potential solutions have been recently visited or diversification rules are violated (48). In terms of the runtime required to find optimal solutions in the regime where exact solutions have been found using branch-and-bound methods (21), Memetic Tabu has been observed to outperform both nonevolutionary methods and memetic algorithms that use simpler neighborhood search schemes such as steepest descent. To verify the scaling of tabu search on the regime of interest for comparison with QAOA, we use the implementation of Memetic Tabu in (20). For each length, we average the runtime over 50 random seeds, obtaining the scaling of the TTS of  $1.35^N$  with a 95% confidence interval of (1.33, 1.38). This scaling closely matches the one reported in (30). We also note that solvers based on self-avoiding random walks

(20) have been shown to be competitive with or outperform Memetic Tabu when the task is to find skew-symmetric sequences with the lowest autocorrelation. These solvers are specialized to search for skew-symmetric sequences and do not naturally extend to the unrestricted LABS problem.

### High-performance simulation of QAOA

Our numerical results are enabled by a custom scalable high-performance algorithm-specific QAOA simulator. We briefly describe the simulator here; for additional details and benchmarks comparing the developed simulator with the state-of-the-art methods for simulating QAOA, the reader is referred to (22).

Here, the main goal of the numerical simulation of QAOA is to evaluate the expectation of the cost Hamiltonian  $\langle C \rangle_{\text{MF}}$  and the success probability  $p^{\text{opt}}$ . Since  $p^{\text{opt}}$  is exponentially small, it has to be evaluated with high precision. While many techniques can be leveraged for exact simulation, we opt to directly simulate the full quantum state as it is propagated through the QAOA circuit. We note in particular that tensor network techniques do not provide a benefit in this case since the circuit we simulate is deep and fully connected [see (22) for detailed comparison].

First, we leverage the observation that the cost Hamiltonian and hence the phase-separation operator are diagonal. This allows us to precompute the cost function evaluated at every binary input and multiply the exponentiated costs elementwise with the statevector to simulate the application of the phase-separation operator. This operation can be easily parallelized since it is an elementwise operation local to each element in the statevector. The same precomputed vector of cost function values is used to compute  $\langle C \rangle_{\text{MF}}$  by taking the inner product with the final QAOA state. The cost of precomputation is amortized over the large number of objective evaluations performed during parameter optimization and is thereby negligible.

Second, we note that the mixing operator consists of an application of a uniform  $x$  rotation applied on each qubit. Therefore, each rotation operation can be computed by multiplying a fixed  $2 \times 2$  unitary matrix with a  $2 \times 2^{n-1}$  matrix constructed from reshaping the statevector. This step is parallelized by grouping the pairs of indices on which the  $2 \times 2$  unitary is applied.

We perform the simulations on the Polaris supercomputer located in Argonne Leadership Computing Facility. We distribute the simulation to 256 Polaris nodes with four NVIDIA A100 GPUs on each node and one AMD EPYC CPU. The CPU is used to manage the communication and the assembly of final results. Each GPU hosts a chunk of the full statevector and a chunk of the integer cost operator vector. Application of the cost operator does not require any communication since it is local to each element. The grouping in the mixing operator depends on index  $i$  of the operator  $x_i$  analogous to the grouping in the fast Walsh-Hadamard transform (49). For  $i \leq n - \log_2(1024) = 29$ , the pairing is local within each GPU. For  $i > 29$ , we use CUDA-enabled MPI to distribute full chunks between nodes, which requires space to be reserved for two statevector chunks on each GPU.

### Supplementary Materials

This PDF file includes:

Notes S1 to S4

Figs. S1 to S19

Table S1

References

## REFERENCES AND NOTES

- C. Dürr, P. Hoyer, A quantum algorithm for finding the minimum. arXiv:quant-ph/9607014 (1996).
- A. Montanaro, Quantum-walk speedup of backtracking algorithms. *Theory Comput.* **14**, 1–24 (2018).
- A. Montanaro, Quantum speedup of branch-and-bound algorithms. *Phys. Rev. Res.* **2**, 013056 (2020).
- S. Chakrabarti, P. Minssen, R. Yalovetzky, M. Pistoia, Universal quantum speedup for branch-and-bound, branch-and-cut, and tree-search algorithms. arXiv:2210.03210 [quant-ph] (2022).
- R. D. Somma, S. Boixo, H. Barnum, E. Knill, Quantum simulations of classical annealing processes. *Phys. Rev. Lett.* **101**, 130504 (2008).
- P. Wocjan, A. Abeyesinghe, Speedup via quantum sampling. *Phys. Rev. A* **78**, 042336 (2008).
- M. B. Hastings, A short path quantum algorithm for exact optimization. *Quantum* **2**, 78 (2018).
- A. M. Dalzell, N. Pancotti, E. T. Campbell, F. G.S.L. Brandão, Mind the gap: Achieving a super-grover quantum speedup by jumping to the end, in *Proceedings of the ACM Symposium on Theory of Computing* (Association for Computing Machinery, 2023), pp. 1131–1144.
- T. Hogg, D. Portnov, Quantum optimization. *Inform. Sci.* **128**, 181–197 (2000).
- E. Farhi, J. Goldstone, S. Gutmann, A quantum approximate optimization algorithm. arXiv:1411.4028 [quant-ph] (2014).
- L. Zhou, S.-T. Wang, S. Choi, H. Pichler, M. D. Lukin, Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. *Phys. Rev. X* **10**, 021067 (2020).
- 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 *Proceedings of the Conference on the Theory of Quantum Computation, Communication and Cryptography* (Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022), pp. 7:1–7:21.
- S. Boulebna, A. Montanaro, Solving Boolean satisfiability problems with the quantum approximate optimization algorithm. arXiv:2208.06909 [quant-ph] (2022).
- S. H. Sureshbabu, D. Herman, R. Shaydulin, J. Basso, S. Chakrabarti, Y. Sun, M. Pistoia, Parameter setting in quantum approximate optimization of weighted problems. arXiv:2305.15201 [quant-ph] (2023).
- A. Boehmer, Binary pulse compression codes. *IEEE Trans. Inform. Theory* **13**, 156–167 (1967).
- M. Schroeder, Synthesis of low-peaked-factor signals and binary sequences with low autocorrelation (Corresp.). *IEEE Trans. Inform. Theory* **16**, 85–89 (1970).
- J. Bernasconi, Low autocorrelation binary sequences : Statistical mechanics and configuration space analysis. *J. Phys. France* **48**, 559–567 (1987).
- S. Mertens, C. Bessenrodt, On the ground states of the Bernasconi model. *J. Phys. A Math. Gen.* **31**, 3731–3749 (1998).
- M. Golay, Sieves for low autocorrelation binary sequences. *IEEE Trans. Inform. Theory* **23**, 43–51 (1977).
- B. Bošković, F. Brzlez, J. Brest, A GitHub archive for solvers and solutions of the LABS problem (2016); [https://github.com/borkob/git\\_labs](https://github.com/borkob/git_labs).
- T. Packebusch, S. Mertens, Low autocorrelation binary sequences. *J. Phys. A Math. Theor.* **49**, 165001 (2016).
- D. Lykov, R. Shaydulin, Y. Sun, Y. Alexeev, M. Pistoia, Fast simulation of high-depth QAOA circuits, in *Proceedings of the SC ’23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis, SC-W 2023* (ACM, 2023), pp. 1443–1451.
- J. E. Gallardo, C. Cotta, A. J. Fernández, Finding low autocorrelation binary sequences with memetic algorithms. *Appl. Soft Comput.* **9**, 1252–1262 (2009).
- J. Lemieux, B. Heim, D. Poulin, K. Svore, M. Troyer, Efficient quantum walk circuits for Metropolis–Hastings algorithm. *Quantum* **4**, 287 (2020).
- S. Boixo, G. Ortiz, R. Somma, Fast quantum methods for optimization. *Eur. Phys. J. Spec. Top.* **224**, 35–49 (2015).
- J. M. Pino, J. M. Dreiling, C. Figgatt, J. P. Gaebler, S. A. Moses, M. S. Allman, C. H. Baldwin, M. Foss-Feig, D. Hayes, K. Mayer, C. Ryan-Anderson, B. Neyenhuis, Demonstration of the trapped-ion quantum CCD computer architecture. *Nature* **592**, 209–213 (2021).
- S. A. Moses, C. H. Baldwin, M. S. Allman, R. Ancona, L. Ascaruzz, 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. Campora III, C. Carron, J. Chambers, J. W. Chan, Y. H. Chen, A. Chernoguzov, E. Chertkov, J. Colina, J. P. Curtis, R. Daniel, M. De Cross, 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. Little, 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. Sprengle, R. P. Stutz, M. Swallows, R. I. Tobey, A. Tran, T. Tran, E. Vogt, C. Volin, J. Walker, A. M. Zolot, J. M. Pino, A race track trapped-ion quantum processor. arXiv:2305.03828 [quant-ph] (2023).
- A. Gonzales, R. Shaydulin, Z. H. Saleem, M. Suchara, Quantum error mitigation by Pauli check sandwiching. *Sci. Rep.* **13**, 2122 (2023).
- D. M. Debroy, K. R. Brown, Extended flag gadgets for low-overhead circuit verification. *Phys. Rev. A* **102**, 052409 (2020).
- B. Bošković, F. Brzlez, J. Brest, Low-autocorrelation binary sequences: On improved merit factors and runtime predictions to achieve them. *Appl. Soft Comput.* **56**, 262–285 (2017).
- Y. R. Sanders, D. W. Berry, P. C. S. Costa, L. W. Tessler, N. Wiebe, C. Gidney, H. Neven, R. Babbush, Compilation of fault-tolerant quantum heuristics for combinatorial optimization. *PRX Quantum* **1**, 020312 (2020).
- J. van Apeldoorn, A. Gilyén, S. Gribling, R. de Wolf, Quantum SDP-solvers: Better upper and lower bounds. *Quantum* **4**, 230 (2020).
- M. Boyer, G. Brassard, P. Hoyer, A. Tapp, Tight bounds on quantum searching. *Fortschr. Phys.* **46**, 493–505 (1998).
- R. Shaydulin, M. Pistoia, QAOA with  $N \cdot p \geq 200$ . arXiv:2303.02064 [quant-ph] (2023).
- Z. He, R. Shaydulin, S. Chakrabarti, D. Herman, C. Li, Y. Sun, M. Pistoia, Alignment between initial state and mixer improves QAOA performance for constrained optimization. arXiv:2305.03857 [quant-ph] (2023).
- E. Pelofske, A. Bärtschi, J. Golden, S. Eidenbenz, High-round QAOA for max k-sat on trapped ion NISQ devices, in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, 2023), pp. 506–517.
- E. Pelofske, A. Bärtschi, S. Eidenbenz, Quantum Annealing Vs. QAOA: 127 qubit higher-order Ising problems on NISQ computers, in *Lecture Notes in Computer Science*. (Springer Nature Switzerland, 2023), pp. 240–258.
- P. Niroula, R. Shaydulin, R. Yalovetzky, P. Minssen, D. Herman, S. Hu, M. Pistoia, Constrained quantum optimization for extractive summarization on a trapped-ion quantum computer. *Sci. Rep.* **12**, 17171 (2022).
- R. Shaydulin, A. Galda, Error mitigation for deep quantum optimization circuits by leveraging problem symmetries, in *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, 2021), pp. 291–300.
- A. Kakkar, J. Larson, A. Galda, R. Shaydulin, Characterizing error mitigation by symmetry verification in QAOA, in *IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, 2022).
- S. Boulebna, A. Montanaro, Predicting parameters for the quantum approximate optimization algorithm for max-cut from the infinite-size limit. arXiv:2110.10685 [quant-ph] (2021).
- J. Basso, D. Gamarnik, S. Mei, L. Zhou, Performance and limitations of the qaoa at constant levels on large sparse hypergraphs and spin glass models, in *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, 2022), pp. 335–343.
- R. Babbush, J. R. McClean, M. Newman, C. Gidney, S. Boixo, H. Neven, Focus beyond quadratic speedups for error-corrected quantum advantage. *PRX Quantum* **2**, 010103 (2021).
- G. Brassard, P. Hoyer, M. Mosca, A. Tapp, Quantum amplitude amplification and estimation. *Contemp. Math.*, 53–74 (2002).
- C. N. Self, M. Benedetti, D. Amaro, Protecting expressive circuits with a quantum error detection code. arXiv:2211.06703 [quant-ph] (2022).
- Gurobi Optimization, [www.gurobi.com](http://www.gurobi.com).
- IBM ILOG CPLEX. V20.1: User’s manual for cplex. International Business Machines Corporation.
- F. Glover, M. Laguna, *Tabu Search* (Kluwer Academic Publishers, 1997).
- F. Algazi, Unified matrix treatment of the fast Walsh–Hadamard transform. *IEEE Trans. Comput.* **C-25**, 1142–1146 (1976).
- M. Golay, A class of finite binary sequences with alternate auto-correlation values equal to zero (Corresp.). *IEEE Trans. Inform. Theory* **18**, 449–450 (1972).
- M. J. E. Golay, D. B. Harris, A new search for skewsymmetric binary sequences with optimal merit factors. *IEEE Trans. Inform. Theory* **36**, 1163–1166 (1990).
- G. F. M. Beenker, T. A. C. M. Claassen, P. W. C. Hermans, Binary sequences with a maximally flat amplitude spectrum. *Philips J. Res.* **40**, 289–304 (1985).
- J. Jedwab, D. J. Katz, K.-U. Schmidt, Advances in the merit factor problem for binary sequences. *J. Comb. Theory Ser. A* **120**, 882–906 (2013).
- M. Golay, The merit factor of long low autocorrelation binary sequences (Corresp.). *IEEE Trans. Inform. Theory* **28**, 543–549 (1982).
- F. F. Ferreira, J. F. Fontanari, P. F. Stadler, Landscape statistics of the low-autocorrelation binary string problem. *J. Phys. A Math. Gen.* **33**, 8635–8647 (2000).
- M. Dimitrov, T. Baitcheva, N. Nikolov, On the generation of long binary sequences with record-breaking psf values. *IEEE Signal Process. Lett.* **27**, 1904–1908 (2020).

57. J. Brest, B. Boskovic, Low autocorrelation binary sequences: Best-known peak sidelobe level values. *IEEE Access* **9**, 67713–67723 (2021).
58. B. Boskovic, J. Brest, Two-phase optimization of binary sequences with low peak sidelobe level value. arXiv:2107.09801 [cs.AI] (2021).
59. J. Lindner, Binary sequences up to length 40 with best possible autocorrelation function. *Electron. Lett.* **11**, 507 (1975).
60. S. Mertens, Exhaustive search for low-autocorrelation binary sequences. *J. Phys. A Math. Gen.* **29**, L473–L481 (1996).
61. F. Brzlez, X.-Y. Li, M. F. Stallmann, B. Militzer, Reliable cost predictions for finding optimal solutions to labs problem: Evolutionary and alternative algorithms, in *Proceedings of The Fifth International Workshop on Frontiers in Evolutionary Algorithms (FEA'2003) under JCIS'2003* (2003).
62. Ivan Dotu, Pascal Van Hentenryck, A note on low autocorrelation binary sequences, in *Principles and Practice of Constraint Programming - CP 2006*, F. Benhamou, Ed. (Springer Berlin Heidelberg, 2006), pp. 685–689.
63. J. Kratica, A mixed integer quadratic programming model for the low autocorrelation binary sequence problem. *Serdica J. Comput.* **6**, 385–400 (2013).
64. S. D. Prestwich, Improved branch-and-bound for low autocorrelation binary sequences. arXiv:1305.6187 [cs.AI] (2013).
65. J. Brest, B. Boskovic, A heuristic algorithm for a low autocorrelation binary sequence problem with odd length and high merit factor. *IEEE Access* **6**, 4127–4134 (2018).
66. J. E. Gallardo, C. Cotta, A. J. Fernandez, A memetic algorithm for the low autocorrelation binary sequence problem, in *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, GECCO07* (ACM, 2007).
67. M. B. Hastings, Classical and quantum bounded depth approximation algorithms. arXiv:1905.07047 [quant-ph] (2019).
68. C.-N. Chou, P. J. Love, J. S. Sandhu, J. Shi, Limitations of local quantum algorithms on random max-k-xor and beyond. arXiv:2108.06049 [quant-ph] (2021).
69. A. Chen, N. Huang, K. Marwaha, Local algorithms and the failure of log-depth quantum advantage on sparse random csps. arXiv:2310.01563 [quant-ph] (2023).
70. R. Shaydulin, I. Safro, J. Larson, Multistart methods for quantum approximate optimization, in *IEEE High Performance Extreme Computing Conference* (IEEE, 2019).
71. G. E. Crooks, Performance of the quantum approximate optimization algorithm on the maximum cut problem. arXiv:1811.08419 [quant-ph] (2018).
72. M. Streif, M. Leib, Training the quantum approximate optimization algorithm without access to a quantum processing unit. *Quantum Sci. Technol.* **5**, 034008 (2020).
73. X. Lee, Y. Saito, D. Cai, N. Asai, Parameters fixing strategy for quantum approximate optimization algorithm, in *International Conference on Quantum Computing and Engineering* (IEEE, 2021).
74. S. H. Sack, M. Serbyn, Quantum annealing initialization of the quantum approximate optimization algorithm. *Quantum* **5**, 491 (2021).
75. O. Amosy, T. Danzig, E. Porat, G. Chechik, A. Makmal, Iterative-free quantum approximate optimization algorithm using neural networks. arXiv:2208.09888 [quant-ph] (2022).
76. J. Larkin, M. Jonsson, D. Justice, G. G. Guerreschi, Evaluation of QAOA based on the approximation ratio of individual samples. *Quantum Sci. Technol.* **7**, 045014 (2022).
77. P. C. Lotshaw, T. S. Humble, R. Herrman, J. Ostrowski, G. Siopsis, Empirical performance bounds for quantum approximate optimization. *Quantum Inf. Process.* **20**, 403 (2021).
78. L. Li, M. Fan, M. Coram, P. Riley, S. Leichenauer, Quantum optimization with a novel Gibbs objective function and ansatz architecture search. *Phys. Rev. Res.* **2**, 023074 (2020).
79. P. K. Barkoutsos, G. Nannicini, A. Robert, I. Tavernelli, S. Woerner, Improving variational quantum optimization using CVaR. *Quantum* **4**, 256 (2020).
80. R. Shaydulin, S. Hadfield, T. Hogg, I. Safro, Classical symmetries and the quantum approximate optimization algorithm. *Quantum Inf. Process.* **20**, 359 (2021).
81. S. Bravyi, A. Kliesch, R. Koenig, E. Tang, Obstacles to variational quantum optimization from symmetry protection. *Phys. Rev. Lett.* **125**, 260505 (2020).
82. Z. Wang, S. Hadfield, Z. Jiang, E. G. Rieffel, Quantum approximate optimization algorithm for MaxCut: A fermionic view. *Phys. Rev. A* **97**, 022304 (2018).
83. S. G. Johnson, The NLopt nonlinear-optimization package (2022). <http://github.com/stevengj/nlopt>.
84. M. J. D. Powell, "The BOBYQA algorithm for bound constrained optimization without derivatives" (Cambridge NA Report NA2009/06, University of Cambridge, 2009).
85. R. Shaydulin, P. C. Lotshaw, J. Larson, J. Ostrowski, T. S. Humble, Parameter transfer for quantum approximate optimization of weighted MaxCut. *Comput. Secur.* **4**, 1–15 (2023).
86. P. C. Lotshaw, G. Siopsis, J. Ostrowski, R. Herrman, R. Alam, S. Powers, T. S. Humble, Approximate Boltzmann distributions in quantum approximate optimization. *Phys. Rev. A* **108**, 042411 (2023).
87. V. Akshay, H. Philathong, E. Campos, D. Rabinovich, I. Zacharov, X.-M. Zhang, J. D. Biamonte, Circuit depth scaling for quantum approximate optimization. *Phys. Rev. A* **106**, 042438 (2022).
88. J. Unger, A. Messinger, B. E. Nieho, M. Fellner, W. Lechner, Low-depth circuit implementation of parity constraints for quantum optimization. arXiv:2211.11287 [quant-ph] (2022).
89. S. Sivarajah, S. Dilkes, A. Cowtan, W. Simmons, A. Edgington, R. Duncan, t|ket>: A retargetable compiler for NISQ devices. *Quantum Sci. Technol.* **6**, 014003 (2021).
90. R. Ayanzadeh, N. Alavizamani, P. Das, M. Qureshi, FrozenQubits: Boosting fidelity of QAOA by skipping hotspot nodes, in *International Conference on Architectural Support for Programming Languages and Operating Systems* (ACM, 2023), pp. 311–324.

**Acknowledgments:** We thank our colleagues at the Global Technology Applied Research center of JPMorgan Chase for support and helpful discussions. Special thanks are also due to T. Uttley and J. Strabley from Quantinuum for their continued support throughout the project. This paper was prepared for informational purposes with contributions from the Global Technology Applied Research center of JPMorgan Chase & Co., Argonne National Laboratory, and Quantinuum LLC. 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 position 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, or financial product or service, or to be used in any way for evaluating the merits of participating in any transaction. The submitted manuscript includes contributions from UChicago Argonne LLC, Operator of Argonne National Laboratory ("Argonne"). Argonne, a US Department of Energy Office of Science laboratory, is operated under contract no. DE-AC02-06CH11357. The US government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (<http://energy.gov/downloads/doe-public-access-plan>). **Funding:** This material is based upon work supported in part by the US Department of Energy, Office of Science, under contract number DE-AC02-06CH11357 and the Office of Science, Office of Advanced Scientific Computing Research, Accelerated Research for Quantum Computing program. **Author contributions:** R.S. devised the project. J.L., N.K., and R.S. implemented QAOA parameter optimization and the parameter setting schemes. R.S. and Y.S. implemented the single-node version of the QAOA simulator. D.L. implemented the distributed version of the QAOA simulator and executed the large-scale simulations on Polaris. R.S. analyzed the simulation results. D.H. and S.H. developed the circuit optimization pipeline. C.L. implemented and analyzed the error detection scheme. M.D. and D.H. executed the experiments on trapped-ion hardware, and C.L. analyzed the results. S.C., D.L., and P.M. benchmarked classical solvers. S.C., D.H., and R.S. analyzed the generalized quantum minimum finding enhanced with QAOA. J.M.D., J.P.G., T.M.G., J.A.G., K.G., D.G., N.H., C.V.H., J.J., M.Ma., T.M., M.Mi., S.A.M., B.N., and P.S. built, optimized, and operated the trapped-ion hardware. M.P. led the overall project. All authors contributed to technical discussions and the writing of the manuscript. **Competing interests:** M.P., R.S., and Y.S. are inventors on a patent application related to this work (no. 18/192,562, filed on 29 March 2023). The authors declare no other competing interests. **Data and materials availability:** All data needed to evaluate the conclusions in the paper are present in the paper and/or the Supplementary Materials. The data presented in this paper can be found at <https://doi.org/10.5281/zenodo.8190275>. The code used to produce the results in this paper can be found at <https://zenodo.org/doi/10.5281/zenodo.10935810>. The latest version can be found at <https://github.com/jpmorganchase/QOKit>.

Submitted 1 November 2023

Accepted 25 April 2024

Published 29 May 2024

10.1126/sciadv.adm6761

# Science Advances

## Evidence of scaling advantage for the quantum approximate optimization algorithm on a classically intractable problem

Ruslan Shaydulin, Changhao Li, Shouvanik Chakrabarti, Matthew DeCross, Dylan Herman, Niraj Kumar, Jeffrey Larson, Danylo Lykov, Pierre Minssen, Yue Sun, Yuri Alexeev, Joan M. Dreiling, John P. Gaebler, Thomas M. Gatterman, Justin A. Gerber, Kevin Gilmore, Dan Gresh, Nathan Hewitt, Chandler V. Horst, Shaohan Hu, Jacob Johansen, Mitchell Matheny, Tanner Mengle, Michael Mills, Steven A. Moses, Brian Neyenhuis, Peter Siegfried, Romina Yalovetzky, and Marco Pistoia

*Sci. Adv.* **10** (22), eadm6761. DOI: 10.1126/sciadv.adm6761

**View the article online**

<https://www.science.org/doi/10.1126/sciadv.adm6761>

**Permissions**

<https://www.science.org/help/reprints-and-permissions>

Downloaded from https://www.science.org on January 31, 2026

Use of this article is subject to the [Terms of service](#)

---

*Science Advances* (ISSN 2375-2548) is published by the American Association for the Advancement of Science. 1200 New York Avenue NW, Washington, DC 20005. The title *Science Advances* is a registered trademark of AAAS.

Copyright © 2024 The Authors, some rights reserved; exclusive licensee American Association for the Advancement of Science. No claim to original U.S. Government Works. Distributed under a Creative Commons Attribution License 4.0 (CC BY).