scientific reports

Check for updates

OPEN

Constrained quantum optimization for extractive summarization on a trapped-ion quantum computer

Pradeep Niroula[1,2,3,4], Ruslan Shaydulin[1,4]✉, Romina Yalovetzky[1,4], Pierre Minssen[1], Dylan Herman[1], Shaohan Hu[1] & Marco Pistoia[1]

Realizing the potential of near-term quantum computers to solve industry-relevant constrained-optimization problems is a promising path to quantum advantage. In this work, we consider the extractive summarization constrained-optimization problem and demonstrate the largest-to-date execution of a quantum optimization algorithm that natively preserves constraints on quantum hardware. We report results with the Quantum Alternating Operator Ansatz algorithm with a Hamming-weight-preserving XY mixer (XY-QAOA) on trapped-ion quantum computer. We successfully execute XY-QAOA circuits that restrict the quantum evolution to the in-constraint subspace, using up to 20 qubits and a two-qubit gate depth of up to 159. We demonstrate the necessity of directly encoding the constraints into the quantum circuit by showing the trade-off between the in-constraint probability and the quality of the solution that is implicit if unconstrained quantum optimization methods are used. We show that this trade-off makes choosing good parameters difficult in general. We compare XY-QAOA to the Layer Variational Quantum Eigensolver algorithm, which has a highly expressive constant-depth circuit, and the Quantum Approximate Optimization Algorithm. We discuss the respective trade-offs of the algorithms and implications for their execution on near-term quantum hardware.

Recent advances in quantum hardware[1–3] open the path for practical applications of quantum algorithms. A particularly promising target application domain is combinatorial optimization. Problems in this space are prominent in many industrial sectors, such as logistics[4], supply-chain design[5], drug discovery[6], and finance[7]. However, many of the most promising quantum algorithms for optimization are heuristic and lack provable performance guarantees. Moreover, limited capabilities of near-term quantum computers further constrain the power of such algorithms to address practically-relevant problems. Therefore, it is crucial to perform thorough evaluations of promising quantum algorithms on state-of-the-art hardware to assess their potential to provide quantum advantage in optimization.

When conducting such evaluations, the choice of the target optimization problem is of particular importance. Many well-studied theoretical problems—such as maximum cut (MaxCut)[8–14], maximum independent set[15] and maximum k-colorable subgraph[16]—have been used to evaluate the performance of quantum optimization algorithms. These problems have several advantages: they are well-characterized theoretically, have strong hardness guarantees, and are easy to map to quantum hardware. At the same time, however, they do not correspond directly to the practically-relevant problems that are solved daily in an industrial setting.

In this work, we consider the problem of *text summarization*, where the goal is to generate a shortened representation of an input document without altering its original meaning. This process is commonly used to produce summaries of news articles[17] and voluminous legal documents[18]. Specifically, we focus on a version of text summarization known as *extractive summarization* (ES), wherein the summary is produced by selecting sentences verbatim from the original text. In our experiments, we map the ES problem to an NP-hard constrained-optimization problem using the formulation introduced by McDonald[19]. The resulting optimization problem is then solved on a quantum computer. We impose a constraint on the number of sentences in the

[1]JPMorgan Chase, New York, NY, USA. [2]Joint Center for Quantum Information and Computer Science, NIST/University of Maryland, College Park, MD, USA. [3]Joint Quantum Institute, University of Maryland, College Park, MD, USA. [4]These authors contributed equally: Pradeep Niroula, Ruslan Shaydulin and Romina Yalovetzky. ✉email: ruslan.shaydulin@jpmchase.com

summary, which is enforced using a penalty term in the objective, or natively by limiting the quantum evolution to a constraint-restricted subspace.

ES is a particularly interesting problem to consider since it has challenges that are similar to those of many other industrially-relevant use cases. First, it is constrained, making it necessary to either restrict the quantum evolution to the corresponding subspace, or introduce large penalty terms into the formulation. Second, it lacks simple structures, such as symmetries[20]. Third, unlike commonly considered toy problems, such as MaxCut, the coefficients in its objective are not necessarily integers, which can make the optimization of quantum algorithm parameters hard[21].

In this paper, we present experimental and numerical results demonstrating the challenges associated with solving constrained-optimization problems with near-term quantum computers. Our contribution is twofold.

First, we demonstrate experimental results showing successful execution of the Quantum Alternating Operator Ansatz algorithm with a Hamming-weight-preserving XY mixer (XY-QAOA)[22,23] on the quantum processor Quantinuum H1-1. We use all 20 qubits of H1-1 and execute circuits with two-qubit gate depth of up to 159 and two-qubit gate count of up to 765, which is the largest such demonstration to date. We additionally report results from the execution of the Layer Variational Quantum Eigensolver (L-VQE)[24], which is a recently-introduced hardware-efficient variational algorithm for optimization. We obtain approximation ratios of up to 92.1% and in-constraint probability of up 91.4% on the H1-1 device.

Second, we motivate our algorithm choice by highlighting the trade-off between the quality of the solution and the in-constraint probability which is implicit in applying unconstrained near-term quantum optimization algorithms to constrained problems. This trade-off suggests the need to carefully engineer the parameter optimization strategy, which is difficult in general. We show how this trade-off can be avoided by either using a sufficiently expressive circuits such as L-VQE (at the cost of the increased difficulty of parameter optimization) or, more naturally, by encoding the constraints directly into the circuits as in the case of XY-QAOA.

The remainder of this paper is organized as follows. Section "Problem description" introduces the quantum algorithms used. Section "Extractive summarization as an optimization problem" describes how the ES problem is formulated as an optimization problem. Section "Methods" details the methodology we adopted to generate the optimization problem instances and solve them on real quantum hardware. Section "Results" illustrates the experimental results we obtained by executing the algorithms and discusses the advantages and downsides of them. Section "Related work" discusses previous hardware demonstrations. Finally, "Discussion" summarizes our findings and their significance. Additional technical details on the algorithms, hyperparameter and ES problem are presented in the Appendix at the end of the paper.

Problem description

For a given objective function f defined on the N-dimensional Boolean cube and a set of feasible solutions $\mathscr{F} \subseteq \{0,1\}^N$, consider the problem of finding a binary string $\mathbf{x} \in \mathscr{F}$ that maximizes it:

$$\max_{\mathbf{x}\in\mathscr{F}} f(\mathbf{x}). \tag{1}$$

The set of feasible solutions $\mathscr{F}$ is typically given by constraints of the form $g(\mathbf{x}) = 0$ or $g(\mathbf{x}) \leq 0$. A binary string $\mathbf{x} \in \mathscr{F}$ is said to be "in-constraint". Let $C \in \mathbb{C}^{2^N \times 2^N}$ denote the Hamiltonian (Hermitian operator) encoding f on qubits. This operator is diagonal in the computational basis ($C = \mathrm{diag}(f(\mathbf{x}))$) and is defined by its action on the computational basis: $C|\mathbf{x}\rangle = f(\mathbf{x})|\mathbf{x}\rangle, \forall \mathbf{x} \in \{0,1\}^N$.

QAOA[25,26] solves the problem (1) by preparing a parameterized quantum state

$$\prod_{j=1}^{p} \left[e^{-i\beta_j \sum_{k=1}^{N} \mathrm{x}_k} e^{-i\gamma_j C} \right] |+\rangle^{\otimes N}, \tag{2}$$

where x_j denotes a single-qubit Pauli x acting on qubit j and the initial state $|+\rangle^{\otimes N}$ is a uniform superposition over all computational basis states. The parameters β, γ are chosen using a classical algorithm, typically an optimization routine[27], with the goal of maximizing the expected objective value of QAOA state-measurement outcomes. The depth of a QAOA circuit is controlled by a free parameter p. In the limit $p \to \infty$, QAOA can solve the problem exactly via adiabatic evolution[25]. Additionally, there exist lower bounds on the performance of QAOA when solving MaxCut in finite depth[12,28], and QAOA achieves performance competitive with best-known classical algorithms on the Sherrington-Kirkpatrick model[29] in infinite-size limit and finite depth[30,31].

In the remainder of this paper, XY-QAOA refers to the Quantum Alternating Operator Ansatz algorithm with a Hamming-weight-preserving XY mixer[22], whereas QAOA indicates the Quantum Approximate Optimization Algorithm[25].

L-VQE[24] solves problem (1) by preparing a parameterized state:

$$\prod_{j=1}^{p} \left[U(\theta_j) \right] V(\theta_0) |0\rangle, \tag{3}$$

where U is a circuit composed of linear nearest-neighbor CNOT gates and single-qubit rotations, V is a tensor product of single-qubit rotations and $|0\rangle$ is the N-qubit vacuum state. Due to the structure of the circuit, the two-qubit gate depth is very low ($4 \times p$ for a circuit with p layers). We refer interested readers to Ref.[24] for the precise definition of the circuit.

While QAOA and L-VQE can tackle constrained-optimization problems in which the constraint is enforced by a penalty term, the output of these algorithms is not guaranteed to satisfy the constraint. Moreover, a penalty

term introduces a trade-off between the in-constraint probability and the quality of the in-constraint solution, as discussed in "Quantum circuit needs to preserve constraints". The Quantum Alternating Operator Ansatz algorithm[22] overcomes this limitation by using a parameterized circuit which limits the quantum evolution to a constraint-preserving subspace. The general form of this circuit is given by

$$\prod_{j=1}^{p} \left[U_M(\beta_j)U_C(\gamma_j)\right]|s\rangle, \tag{4}$$

where U_C is the *phase operator*, which encodes the objective and is diagonal in the computational basis, U_M is a non-diagonal *mixer operator*, and $|s\rangle$ is some initial state. QAOA circuit can be recovered as a special case by setting $U_M(\beta_j) = e^{-i\beta_j \sum_{k=1}^{N} \mathrm{x}_k}$, $U_C(\gamma_j) = e^{-i\gamma_j C}$ and $|s\rangle = |+\rangle^{\otimes N}$. In this paper, we focus in particular on the Hamming-weight constraint of the form $\sum_{j=1}^{N} x_j = M$, which in our case corresponds to fixing the size of the summary. While many variations of this algorithm exist[22], we only consider the XY-QAOA version, which lends itself well to implementation on near-term hardware. We let the initial state be the uniform superposition over all binary strings with Hamming weight M, $U_C(\gamma_j) = e^{-i\gamma_j C}$ and $U_M^{\mathrm{xy}}(\beta_j) = \prod_{k=1}^{N} e^{-i\frac{\beta_j}{2}(\mathrm{x}_k\mathrm{x}_{k+1}+\mathrm{y}_k\mathrm{y}_{k+1})}$[23]. Given that $\sum_{k=1}^{N} \mathrm{x}_k\mathrm{x}_{k+1} + \mathrm{y}_k\mathrm{y}_{k+1}$ commutes with $\sum_{k=1}^{N} \mathrm{z}_i$, the evolution produced by the resulting circuit is restricted to the span of computational basis states with Hamming weight M, as desired.

Extractive summarization as an optimization problem. Extractive summarization (ES) is an interesting problem to evaluate the performance of quantum optimization algorithms due to its practical importance and complex structure. The goal of ES is to pick a subset of the sentences present in a large document to form a smaller document such that this subset preserves the information content in the original document, i.e., summarizes it. While many approaches to solving ES exist (see e.g.[32–35]), in this work we focus on a particular formulation of ES as an optimization problem. Specifically, we consider the problem of *maximizing the centrality* and *minimizing the redundancy* of the sentences in the summary under the constraint that the total number of sentences in the summary is fixed. This formulation of ES has been proposed by R. McDonald and shown to be NP-hard to solve exactly[19].

We now introduce the necessary notation and formally define the problem. An ES algorithm maps a document of N sentences to a summary of $M < N$ sentences. Let the sentences be denoted by integers $i \in [N] := \{0, 1, \ldots N-1\}$ according to the order in which they appear in the document. An extractive summary is a vector $\mathbf{x} \in \{0,1\}^N$, where $x_i = 1$ if and only if sentence i is included in the summary text associated with $\mathbf{x}$. The summary should identify sentences that are central, meaning important, to the document. The salience of a sentence is measured with some centrality which is a map $\mu : [N] \to \mathbb{R}$ satisfying the following property: $\mu(i) > \mu(j)$ if and only if sentence i contains more information about the document than j. At the same time, to keep the summary short, it is desirable to ensure that the sentences in the summary are not redundant. The overlap in information content between two sentences is measured with *pairwise similarity* which is a symmetric map $\beta : [N] \times [N] \to \mathbb{R}$ that satisfies the following property: $\beta(i,j) > \beta(i,k)$ if and only if sentence j is more similar to i than k is to sentence i. We discuss the particular choices of measures of sentence centrality and pairwise similarity in Appendix 1.

The extractive summarization is formulated as the following optimization problem[19]:

$$\begin{aligned} \max_{\mathbf{x}\in\{0,1\}^N} & \sum_{i=0}^{N-1} \mu(i)x_i - \lambda \sum_{i\neq j} \beta(i,j)x_ix_j, \\ \text{s.t.} & \sum_{i=0}^{N-1} x_i = M \end{aligned} \tag{5}$$

where the parameter λ controls how the inclusion of similar sentences in the summary is penalized. The objective of this maximization problem is to increase the information content of the sentences in the summary while ensuring that the total pairwise similarity between sentences is low, relative to λ. Refer to Appendix 5 that shows how this parameter affects the quality of the summaries obtained.

This problem can be solved directly by a quantum algorithm that can preserve constraints. On the other hand, to solve this problem using an unconstrained-optimization algorithm, we must convert this problem to an unconstrained one by adding a penalty term to enforce the constraint. The penalty term is minimized when exactly M sentences are selected. This term is weighted with the parameter Γ. Including the constraint on the lengths of summaries, the optimization problem becomes the following:

$$\max_{\mathbf{x}\in\{0,1\}^N} \sum_{i=0}^{N-1} \mu(i)x_i - \lambda \sum_{i\neq j} \beta(i,j)x_ix_j - \Gamma\left(\sum_{i=0}^{N-1} x_i - M\right)^2, \tag{6}$$

which can be simplified further by ignoring constant terms to get a quadratic objective:

$$\max_{\mathbf{x}\in\{0,1\}^N} \sum_{i=0}^{N-1} \underbrace{(\mu(i) + 2\Gamma M - \Gamma)}_{\mu_{\Gamma,M}(i)} x_i - \sum_{i\neq j} \underbrace{(\lambda\beta(i,j) + \Gamma)}_{\beta_\Gamma(i,j)} x_i x_j. \tag{7}$$

Methods

To generate the optimization problems to be solved, we use articles from the CNN/DailyMail dataset[36]. This dataset contains just over 300 k unique news articles written by journalists at CNN and the Daily Mail in English. The optimization-problem instances consist of two sets of 10 instances: one set with $N = 20$ and a required summary length of $M = 8$ and another set with $N = 14$ and a required summary length of $M = 8$. We use sentence embeddings produced by BERT[37] to compute similarities, and tf-idf[38] to compute centralities, both of which are discussed in detail in Appendix 1.

To quantify the quality of the solution for the optimization problem, we report the approximation ratio given by

$$\frac{f_{\text{observed}} - f_{\min}}{f_{\max} - f_{\min}}, \tag{8}$$

where f_{observed} is the objective value for the solution produced by a given algorithm, $f_{\max}$ is the maximum value of the objective function and $f_{\min}$ the minimum. The maximum and minimum are computed over all possible in-constraint solutions. For quantum algorithms, f_{observed} is computed as the average value of the objective function over all in-constraint samples. We additionally report the probability of the solution being in-constraint, which for the experimental results is estimated as the ratio of in-constraint samples to the total number of samples. In practice, the running time of the algorithm scales inversely proportionally to the in-constraint probability due to the need to obtain at least one in-constraint sample.

For unconstrained quantum optimization solvers, the cost function to maximize contains a penalty term with weight Γ to constrain the number of sentences in the summary to be M (last term in (6)). The value of this hyperparameter Γ was chosen to ensure the value of the cost in (6) corresponding to in-constraint binary strings is greater than the cost corresponding to out-of-constraint binary strings. For each article, after having calculated the similarity and centrality measures, we set $\Gamma = \sum_{i=0}^{N-1} \mu(i) + \lambda \sum_{i\neq j} \beta(i,j)$, where $\lambda = 0.075$. See Appendix 5 for additional numerical experiments showing the impact of the value of λ on the performance.

We execute the QAOA, XY-QAOA and L-VQE using 14 and 20 qubits respectively. We use one layer for QAOA and XY-QAOA ($p = 1$) and for L-VQE we use $p = 1$ for 14-qubit problems and $p = 2$ for 20-qubit problems. We optimize the parameters in noiseless simulation and then execute the circuits with optimized parameters on hardware with 2000 shots. We use Qiskit[39] for circuit manipulation and noiseless simulation. For the hardware experiments, we transpile and optimize the circuits to H1-1's native gate set using Quantinuum's t|ket⟩ transpiler[40]. For comparison, we also run the quantum algorithms on an emulator provided by Quantinuum, which approximates the noise of H1-1.

For the XY-QAOA circuit, a significant part of the two-qubit gate depth comes from the circuit preparing the initial state, which is a uniform superposition of all in-constraint states (Dicke state). To obtain circuits that are shallow enough to be executable on hardware, we leverage recently developed techniques for the short-depth Dicke-state preparation[41–43]. Specifically, we use the divide-and-conquer approach[43] to generate circuits targeting a device with an all-to-all connectivity.

Results

We now present experimental results obtained in simulation and on the Quantinuum H1-1 quantum processor. We show the largest to date demonstration of constrained quantum optimization on gate-based quantum computers, using up to 20 qubits and 765 native two-qubit gates (see "Related work" for a review of previous demonstrations). Note that in the results presented below, we do not perform any error mitigation for the results obtained from hardware or noisy simulations. The specifications of the H1-1 processor are given in Appendix 3.

As previously discussed, we use three quantum optimization algorithms to solve a constrained-optimization problem with the eventual goal of generating document summaries. The optimization algorithms we use are QAOA, L-VQE and XY-QAOA. See "Problem description" for the definitions and discussion of the algorithms, and Appendix 2 for the implementation details. All the statistics we report are computed over 10 problem instances for each number of qubits, with the exception of XY-QAOA on H1-1 where only three 14-qubit and three 20-qubit instances are solved due to the high circuit depth and correspondingly high running time on trapped-ion hardware. "Random" and "Random in-constraint" always refers to statistics computed over all binary strings and all in-constraint binary strings respectively. This is equivalent to computing them with respect to uniform random distribution over corresponding sets.

Experiments on hardware. In Fig. 1, we present the approximation ratios and the in-constraint probabilities for the three algorithms obtained from the execution in a noiseless simulator, with approximated noise in the emulator of H1-1 and on the real H1-1 device. For comparison, we also present the expected approximation ratio of a random feasible solution. We observe that all the approximation ratios obtained, including those from largest circuit executions on hardware, significantly improve upon random guess. Additionally, we observe that the results on hardware are on average at least as good as those obtained from the emulator, making us confident that the emulator gives a good lower bound on the solution quality that can be expected on hardware.

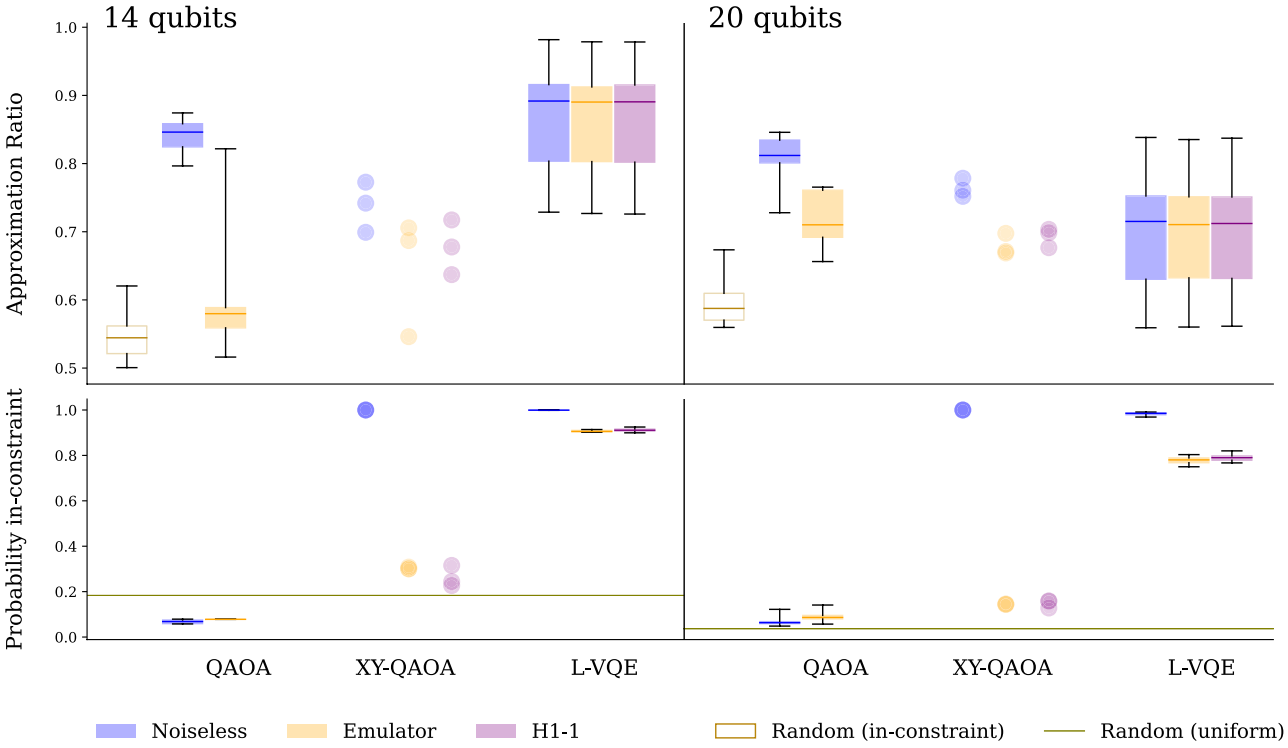

Figure 1. The approximation ratios (top) and in-constraint probabilities (bottom) obtained in the optimization of the instances on 14 (left) and 20 (right) qubits using different quantum optimization algorithms executed in a noiseless simulator, the Quantinuum H1-1 machine and its emulator. In each box plot, the box shows quartiles, the line is the median and the whiskers show minimum and maximum. Each plot is showing the statistics over 10 problem instances, with the exception of XY-QAOA where only three 14-qubit and three 20-qubit runs have been executed due to high circuit depth. We observe that all results significantly improve on random guess. The in-constraint probability of QAOA is below random guess for 14 qubits due to the choice of parameters; see discussion in "Quantum circuit needs to preserve constraints".

Examining Fig. 1 makes apparent the relative advantages and limitations of the three algorithms we consider. We begin by noting that QAOA gives a relatively high approximation ratio, but a very low in-constraint probability. This is due to the QAOA parameters being chosen to trade-off the two metrics of success; we discuss this issue in detail in "Quantum circuit needs to preserve constraints" below. Here we simply note that the low in-constraint probability of QAOA makes the high approximation ratio less significant. Combined with the complexity of parameter setting arising from the trade-off, this means that QAOA is not a good algorithm for the problem we consider here despite having a relatively high approximation ratio. This trade-off is avoided by L-VQE and XY-QAOA in two different ways.

L-VQE uses a very expressive parameterized circuit that can in principle solve the problem exactly with no two-qubit gates, just by optimizing the parameters of the initial single-qubit-gate layer $V(\theta_0)$. At the same time, the expressiveness of L-VQE circuit makes the parameters hard to optimize, both due to their high number and due to the gradients vanishing as the number of qubits grows in some cases[44,45]. As we consider modestly sized problems in this work, we are able to optimize the parameters and obtain solutions with very high approximation ratio and in-constraint probability. However, as the number of qubits grows, this will become increasingly infeasible.

XY-QAOA natively encodes the constraints by restricting the quantum evolution to the subspace equal to the span of the computational basis states of Hamming weight M. This leads to an in-constraint probability of one and a high approximation ratio, if no noise is present. At the same time, XY-QAOA requires deeper circuits as compared to QAOA and L-VQE. This is due to the higher gate count cost of the XY-QAOA mixer operator and the initial state preparation. The two-qubit gate counts and two-qubit gate depths of the executed circuits are given in Table 1.

The unconstrained QAOA uses a mixer unitary which is a product of single-qubit rotations, i.e., $U_M(\beta_j) = \prod_{k=1}^{N} e^{-i\beta_j x_k}$, where N is the number of qubits. For most near-term circuit architectures, including H1-1, the single qubit gates are relatively less noisy than the two-qubit entangling gates[46,47]. As a result, the mixer unitary does not add much noise to the evolution. Each time-step of QAOA, therefore, has at most $N(N-1)/2$ entangling gates, all of which come from the pairwise interaction in the problem Hamiltonian. On the other hand, XY-QAOA uses a more complex mixer operator, which preserves the Hamming weight of the states it acts on. This operator is defined as: $U_M^{XY}(\beta_j) = \prod_{k=1}^{N} e^{-i\frac{\beta_j}{2}(x_k x_{k+1} + y_k y_{k+1})}$. It adds additional $O(N)$ gates in each layer of QAOA circuit as it requires entangling gates on all adjacent pairs.

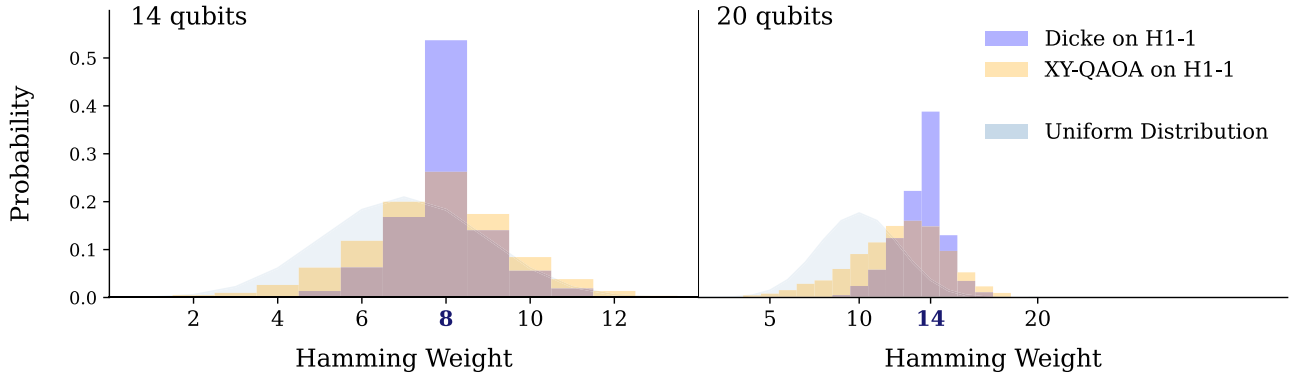

Figure 2. Hamming weight of the bitstrings sampled from the initial uniform superposition of in-constraint states (Dicke state) and a full XY-QAOA circuit. The in-constraint Hamming weight is highlighted in bold. Unlike the QAOA or L-VQE, in XY-QAOA a significant amount of noise is incurred in the initial state preparation step. Observe that the initial Dicke state, due to hardware noise, includes out-of-constraint states.

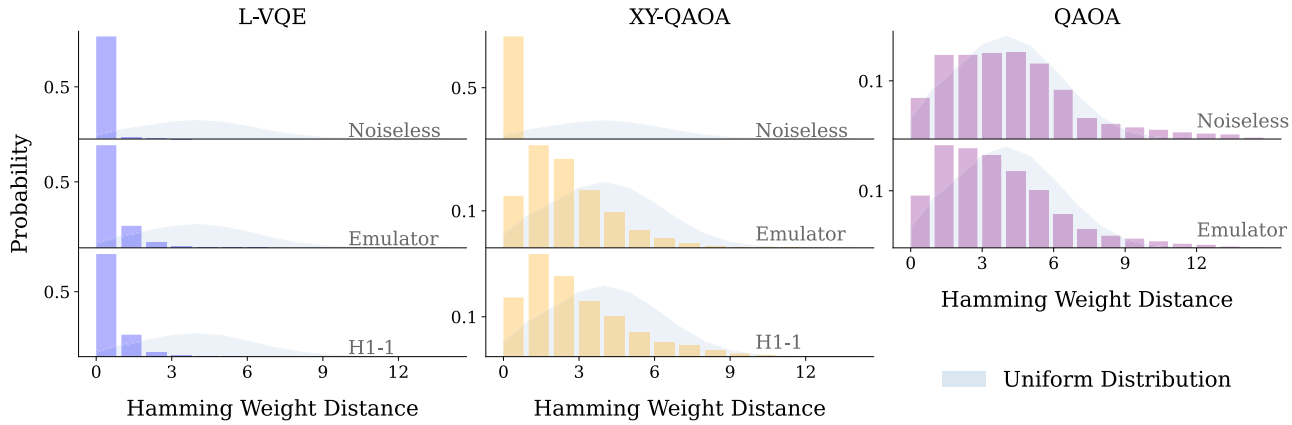

Figure 3. The probability of sampling bitstrings with a given Hamming weight distance to the in-constraint subspace for 20 qubits. The Hamming weight distance is given by $|\mathrm{wt}(x) - M|$, where $\mathrm{wt}(x)$ is the Hamming weight of x and M is the constraint value. The shaded background represents the distribution of Hamming weight distance for random bitstrings sampled from a uniform distribution. Note that the overlap with the in-constraint subspace is lower on hardware due to the presence of noise.

The implementation of XY-QAOA requires preparing an initial state, which is a uniform superposition of all in-constraint states (Dicke state). This state is non-trivial to implement, as it requires a circuit with $O(N)$ two-qubit gate depth[41]. This is in sharp contrast with the initial state used by QAOA or L-VQE, which requires no two-qubit gates to prepare. Therefore in XY-QAOA a significant cost is incurred before any optimization is performed. Figure 2 shows this by plotting the probabilities of sampling bitstrings with different Hamming weights from the output of the initial state preparation circuit and the full XY-QAOA circuit on the H1-1 device. Unlike the noiseless case, the initial state is not fully contained in the in-constraint subspace. At the same time, the noise is sufficiently low so that the XY-QAOA output distribution is still concentrated on the in-constraint subspace. Improved hardware fidelity would lead to more accurate initial state preparation and higher overall in-constraint probability.

Finally, we examine the impact of noise on the in-constraint probability of the output of all three algorithms. As discussed above, for our problem of text summarization, it is crucial that the Hamming weight of the output solution is constrained. Both the algorithm design and the hardware noise affect the in-constraint probability. In Fig. 3, we visualize this behavior for 20-qubit instances, with the analogous figure for 14 qubits given in Appendix 2. Concretely, we plot the probability of obtaining a bitstring $\boldsymbol{x}$ with Hamming distance k to in-constraint bitstrings for each $k \in \{0, \ldots, N - M\}$. Hamming distance k means that at least k bitflips are required to transform one bitstring into the other. We can see that as the noise is added, the in-constraint probability decreases and the output begins to include out-of-constraint bitstrings. Note that for the short L-VQE circuits, the amount of noise accumulated during the circuit execution is small, and only the bitstrings that are one or two bitflips away are included in the output. On the other hand, for deeper XY-QAOA circuits, bitstrings as far as 10 bitflips away are included. For QAOA, even the noiseless output includes primarily out-of-constraint bitstrings due to the choice of parameters.

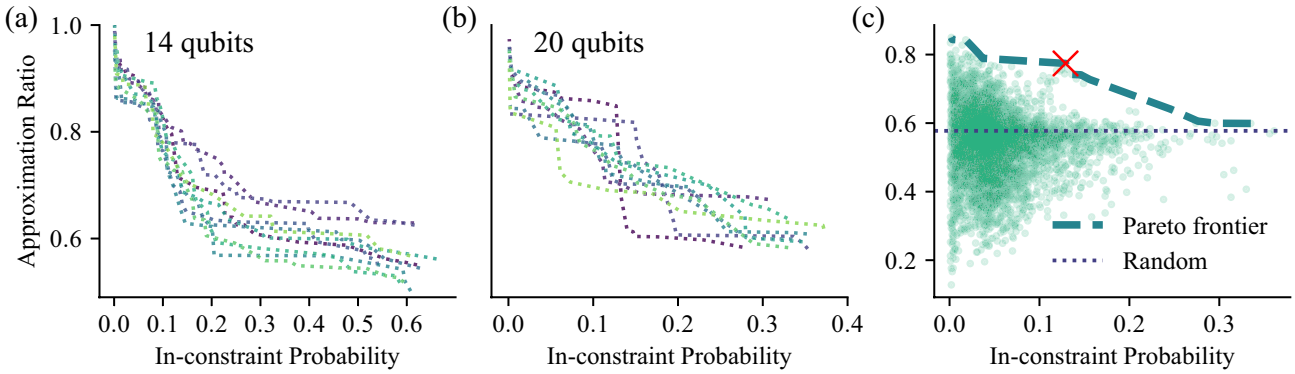

Figure 4. Pareto frontier of QAOA parameters with respect to approximation ratio and the in-constraint probability for 14 (**a**) and 20 (**b**) qubits. Each color plots a different problem instance for a given number of qubits. When solving a constrained-optimization problem using QAOA, the parameter optimization strategy has to be tuned to arrive at the desired point on the Pareto frontier. Directly optimizing the objective (6) would lead to choosing the rightmost point of the frontier, which in the case of the 20 qubit instance presented in (**c**) corresponds to approximation ratio equal to that of random guess. In (**c**), each dot corresponds to one value of β, γ from a grid search in parameter space. Thick dashed line shows the Pareto frontier and the thin dotted line marks the approximation ratio of a random solution. The red $\times$ marker indicates the parameters chosen for the experiments shown above.

Algorithm	# qubits	Initial State		Full Circuit	
		2Q depth	2Q count	2Q depth	2Q count
L-VQE	14	0	0	4	26
	20	0	0	8	76
QAOA	14	0	0	50	182
	20	0	0	74	380
XY-QAOA	14	68	185	114	393
	20	95	347	159	765

Table 1. Two-qubit gate depths (2Q depth) and two-qubit gate counts (2Q count) of the circuits executed on emulator and on the H1-1 device. The circuits were optimized using pytket[40], and the optimized numbers are reported. Initial state preparation for QAOA and L-VQE requires no two-qubit gates.

Quantum circuit needs to preserve constraints. As discussed above, the output distribution of QAOA circuits has a relatively small overlap with the in-constraint space even in the absence of noise. This is due to the trade-off between the in-constraint probability and the approximation ratio, which is implicit in the choice of the parameter optimization strategy. This trade-off is an important weakness of unconstrained quantum algorithms applied to constrained problems, necessitating the development and implementation of quantum algorithms that natively preserve constraints. In the QAOA experiments presented in Fig. 1, we choose the parameters with the goal of increasing the approximation ratio of in-constraint solutions at the expense of reduced in-constraint probability. We now examine this trade-off numerically.

We perform a grid search over the parameter space β, γ of a single-layer QAOA. In Fig. 4 we plot the Pareto frontier with respect to approximation ratio and in-constraint probability. Specifically, we plot the approximation ratios and in-constraint probabilities for QAOA with parameter values β, γ such that there do not exist parameters $\hat{\beta}, \hat{\gamma}$, which improve either of the metrics without decreasing the other. We plot such frontiers for all 20 instances considered.

This trade-off behavior is not specific to QAOA and indeed applies to all unconstrained quantum algorithms that are not expressive enough to solve the problem exactly. Examining the Pareto frontiers makes clear the challenge of choosing parameters for such algorithms. Whereas in this work we perform the full grid search and we may actually choose any point on the frontier for execution on hardware, in practice an objective function must be carefully designed to optimally trade-off the two objectives. This is hard in general. As an example, optimizing parameters with respect to (6) would lead to prioritizing the in-constraint probability. Figure 4c shows an example of an instance where this leads to approximation ratio equal to that of random guess.

One potential solution to this challenge is using a circuit that is sufficiently expressive to solve the problem exactly, such as the circuit used in L-VQE. However, such circuits by necessity would have many parameters that are hard to optimize. Additionally, in many cases they would suffer from gradients of the objective vanishing exponentially with the number of qubits ("barren plateaus")[45], making optimization impossible. Therefore the most practical solution is encoding constraints directly into the quantum circuit, as is the case for XY-QAOA.

	References	Year	Hardware	Connectivity	N	p	2q-gate depth	Error Mit.	H.W.C.
Unconstrained	[48]	2017	Superconducting	Nearest neighbor	19	1	6	No	–
	[49]	2020	Superconducting	Nearest neighbor	7	6	42	No	–
	[13]	2021	Superconducting	Nearest neighbor	23	5	40[1]	Yes	–
	[13]	2021	Superconducting	Nearest neighbor	17	3	153	Yes	–
Hard constraint	[50]	2022	Superconducting	Nearest neighbor	3	5	–	No	2
	[50]	2022	Trapped ion	All-to-all	3	4	–	No	2
	This work	2022	Trapped ion	All-to-all	14	1	114	No	8
	This work	2022	Trapped ion	All-to-all	20	1	159	No	14

Table 2. Comparison of the hardware demonstrations shown in this work with previous hardware demonstrations of QAOA on gate-based devices. We only report experiments that outperformed random guess and either are similar in scale to ours in terms of circuit size or utilized hard constraints. All hard constraints are on the Hamming weight (H.W.C. in the table). The two-qubit-gate depth is missing from references where the depth was not reported and could not be estimated. We also report the topology of two-qubit interactions available o the hardware ("Connectivity" in the table) and whether error-mitigation techniques were applied ("Error Mit."). Depth estimated from the circuit description in the paper.

Related work

In this work, we show the largest demonstration to date of constrained optimization on a gate-based quantum computer. We now briefly review the state-of-the-art, which we summarize in Table 2. We include the unconstrained-optimization demonstrations for completeness, and emphasize that the circuits used in this work are deeper than any quantum optimization circuits executed previously for any problem.

There have been various quantum hardware demonstrations of QAOA applied to unconstrained-optimization problems. Using Google's "Sycamore" superconducting processor, Harrigan et al. [13] ran QAOA to find the ground states of Ising models that mapped to the 23-qubit Sycamore topology and Sherrington–Kirkpatrick models with 17 vertices. Additionally, they solved MaxCut on 3-regular graphs with up to 22 vertices. Otterbach et al. solved MaxCut on a 19-vertex graph that obeys the hardware topology of Rigetti's 19-qubit "Acorn" superconducting processor[48]. Lacroix et al. [49] executed QAOA, up to $p = 6$, on a superconducting gate-based quantum computer to solve the exact-cover problem on at most seven vertices. The device supported two-qubit controlled arbitrary-phase gates. This enhanced gate set resulted in a two-qubit-gate depth of 42. There have been other hardware demonstrations of solving unconstrained-optimization problems, with either lower qubit counts or shallower circuits[14,51–59].

An important step in implementing XY-QAOA is the preparation of Dicke states. Aktar et al. [43], who developed the divide-and-conquer Dicke-state preparation approach used in our experiments, implemented their algorithm using up to six qubits on two IBM Q devices.

To the authors' knowledge, there has only be one other demonstration of QAOA with constrained mixers and Dicke-state initialization on quantum hardware. This was done by Baker and Radha[50] who solved problems using at most five qubits and $p = 5$. They executed circuits using both the XY complete-graph mixer with Dicke-state initialization and the XY ring mixer with initialization to a random in-constraint state. They utilized the Rigetti "Aspen-10" superconducting processor, five IBM Q superconducting processors, and the 11-qubit IonQ trapped-ion device. They reported that QAOA beat, with both the XY complete-graph mixer and XY ring mixer, random guess for up to three qubits and $p = 5$ on the Rigetti device. Lastly, their results on IonQ, for the XY ring mixer, beat random guess for up to three qubits and $p = 4$.

While there have been other hardware demonstrations of QAOA using alternative mixers to the transverse field, they were either not applied to hard-constrained problems or did not use an in-constraint initial state. For example, Golden et al. [60] solved six Ising problems on 10 IBM Q backends using QAOA with Grover mixers[61]. While Grover mixers can be used to incorporate hard constraints, the problems Golden et al. solved were unconstrained. Pelofske et al. [62] solved five Ising problems, still unconstrained, with QAOA using Grover mixers on seven IBM Q backends, Rigetti's "Aspen-9" device, and the 11-qubit IonQ device. The instances required fewer than seven qubits. For a protein folding problem, Fingerhuth et al. [63] executed QAOA with an XY mixer using four qubits and $p = 1$ on Rigetti's Acorn device. However, the initial state was a uniform superposition over all bitstrings, and thus the Hamming weight constraint was not obeyed.

Lastly, there have been very large-scale demonstrations of QAOA on analog quantum simulators. First, Pagano et al. [64] demonstrated QAOA on a trapped-ion analog quantum simulator using up to 40 qubits at $p = 1$ and 20 qubits at $p = 2$ on unconstrained problems. Second Ebadi et al. [15] show an application of an analog quantum simulator, using QAOA, to maximum independent set (MIS) problems on graphs with up to 179 vertices. This variant of QAOA was performed by controlling the timing of global laser pulses applied to a 2D array of 289 cold atoms. The global pulses induce Rydberg excitations, which result in a blockade effect that ensures that only independent sets are sampled. However, since these analog devices do not implement universal gate sets, the results are not directly comparable to ours.

Discussion

In this work, we present the largest to date demonstration of constrained optimization on quantum hardware. Our results demonstrate how algorithmic and hardware progress are bringing the prospect of quantum advantage in constrained optimization closer, which can be leveraged in many industries including finance[65,66].

In our experiments on the 20-qubit Quantinuum H1-1 system, we observe that XY-QAOA with up to 20 qubits provides results that are significantly better than random guess despite the circuit depth exceeding 100 two-qubit gates. This progress can be clearly observed by comparing the size and the complexity of the circuits used in our experiments to the previous results discussed in "Related work" above. The results we present here were obtained with no error mitigation, which is not the case for many of the previous demonstrations. Our execution of complex circuits for constrained optimization benefits from the underlying hardware's all-to-all connectivity, as the circuit depth would increase significantly if the circuit had to be compiled to a nearest-neighbor architecture.

We additionally show the necessity of embedding the constraints directly into the quantum circuit being used. If the circuit does not preserve the constraints, the in-constraint probability and the quality of the in-constraint solution have to be traded-off against each other. This trade-off is hard to do in general. This observation further motivates our investigation of XY-QAOA on H1-1, and gives additional weight to our results. At the same time, we show that further advances are needed to reduce the hardware requirements of implementing such circuits and improve the fidelities of the hardware.

Data availability

The data presented in the paper is available at https://doi.org/10.5281/zenodo.6819861.

Appendix 1: Details of the optimization problem formulation

The formulation of extractive summarization as an optimization problem requires a meaningful measure of centrality of sentences and similarity between two sentences. To calculate similarity, we use vector embeddings obtained from a neural language model[37], specifically the Bidirectional Encoder Representations from Transformers (BERT)[67]. Neural Transformers[68] have resulted in many recent successes in various natural language processing tasks[69]. We measure sentence similarity by computing the *cosine similarity* of BERT embeddings defined as

$$\frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\|_2 \|\mathbf{v}_j\|_2}, \tag{9}$$

where $\mathbf{v}_j \cdot \mathbf{v}_i$ denotes the usual dot product, and $\|\cdot\|_2$ the ℓ_2 norm. We refer the reader to the work done by Achananuparp et al. [70] for various techniques used to measure similarity.

For a measure of centrality, we use the tf-idf statistic[38,71]. In contrast to embeddings obtained from pre-trained language models like BERT, *term frequency-inverse document frequency* (tf-idf) is a technique of measuring the "salience" or "importance" of a word in a document by counting its occurence in the entire corpus. The tf-idf measure of a word is computed by multiplying a measure of term frequency (tf) with a measure of inverse document frequency (idf). The tf is the proportion of a document, D, that contains the word w. The idf scales down the raw proportion based on the how frequently the word occurs across all documents. Mathematically, for a word w appearing in a sentence s of a document D,

$$\text{tf-idf}(w,S,D) := \underbrace{\left(\frac{f_{w,S}}{\sum_{w' \in S} f_{w',S}}\right)}_{\text{tf(w,S)}} \times \underbrace{\left(\log \frac{N}{|\{S \in D : w \in S\}|}\right)}_{\text{idf(w,D)}}, \tag{10}$$

where $f_{w,S}$ is the frequency of word w in S and N is the number of unique words in the document.

While the measure in (10) was defined for words, we need a similar measure defined for sentences. To define a tf-idf-based measure for sentences, we take the mean of the tf-idf values of the words in the sentence:

$$\overline{\text{tf-idf}}(S) := \frac{1}{n_S} \sum_{w \in S} \text{tf-idf}(w,S,D) \tag{11}$$

where n_S is the length of the sentence S.

Appendix 2: Details of numerical studies

For each instance of the optimization problem, we evaluate QAOA for each of the values in the grid using 1000 shots. In order to decide the best parameters for each optimization instance, we impose a threshold on the QAOA in-constraint ratio to be higher than 0.06 and we selected the γ, β that maximize the expected approximation ratio. For all the instances, the value obtained was higher than the expected approximation ratio of a random feasible solution.

For L-VQE and XY-QAOA the parameters are optimized by running COBYLA[72,73] from a fixed number of randomly chosen initial points. The number of initial points is 20 for L-VQE with 14 qubits, 5 for L-VQE with 5 qubits and 10 for XY-QAOA for all qubit counts. We choose the parameters giving the best approximation ratio. n. In the main text, we present the distribution of Hamming weight distances for optimization instances on 20 qubits. For completeness, we present the same distribution for 14 qubits in Fig. 5. As it happens with 20 qubits, for algorithms with shallow circuits (L-VQE) the ideal distribution is concentrated around zero even when executed on hardware. Whereas XY-QAOA, an algorithm that natively preserve cardinality, when executed on noiseless simulation produces a distribution concentrated at $k = 0$, when it is executed on the emulator and real

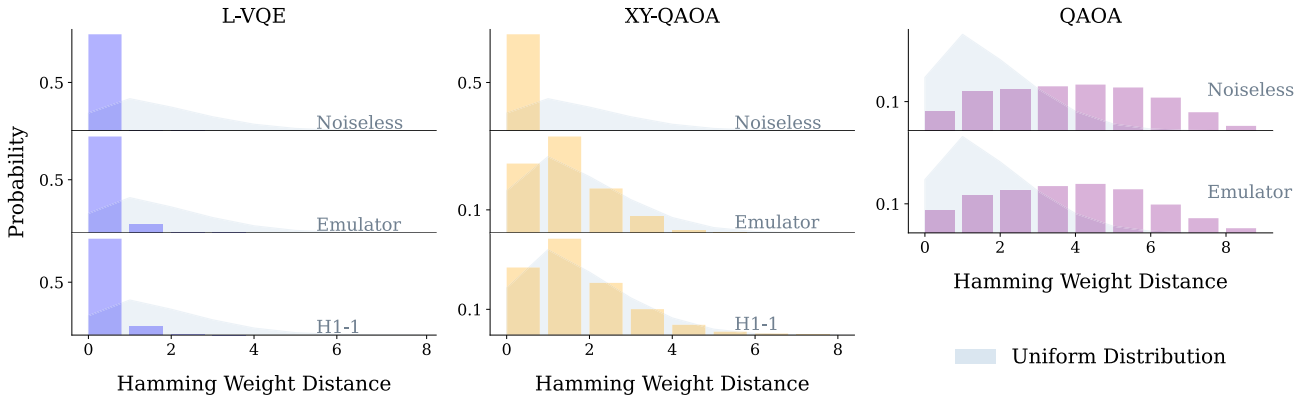

Figure 5. The probability of sampling bitstrings with a given Hamming weight distance to the in-constraint subspace for 14 qubits. The Hamming weight distance is given by $|\mathrm{wt}(x) - M|$, where wt (x) is the Hamming weight of x and M is the constraint value. The shaded background represents the distribution of Hamming weight distance for random bitstrings sampled from a uniform distribution. Note that the overlap with the in-constraint subspace is lower on hardware due to the presence of noise. For QAOA, the in-constraint probability is lower than that of random guess due to parameter choice as we fix the target in-constraint probability to be 0.06.

hardware, it produces a much wider distribution and it is more similar to the random distribution. Notably, for QAOA, both in noiseless simulator and in emulator, the distribution is concentrated far away from $k = 0$ and it is even concentrated at a k higher that the center of the random distribution.

Appendix 3: Details of the H1-1 quantum processor

The Quantinuum H1-1 Quantum Processor uses quantum charge-couple device architecture with five parallel gate zones in a linear trap. The quantum states are stored in hyperfine states of twenty $^{171}Yb^{+}$ atoms. All-to-all connectivity is implemented by rearranging of the physical location of qubits, which introduces a negligible amount of error. Typical single-qubit gate infidelity is 5×10^{-5} and typical two-qubit gate infidelity is 3×10^{-3}. Typical error rate of state preparation and measurement is 3×10^{-3}. Memory error per qubit at average depth-1 circuit ("idle error") is 4×10^{-4}. Additional details are available in Ref.[74]

Appendix 4: Evaluation of optimization-generated summaries

A popular evaluation metric used for summarization tasks is the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metric[75]. We calculate F1 score of the three variants of ROUGE, namely ROUGE-1-F, ROUGE-2-F and ROUGE-L-F, which measure unigram-similarity, bigram similarity and longest-common subsequence, respectively.

Each of the optimization instances discussed in the main text corresponds to an article from the CNN/DailyMail dataset, consisting of 14 or 20 sentences, with the linear and quadratic coefficients in the optimization objective generated using centrality and similarity measures discussed in Sec A. The predicted summaries consist of 8 and 14 sentences respectively. For each of the optimization instances, we calculate the ROUGE metrics with the output distribution of the quantum algorithms, weighted by the probability of measuring an in-constraint bitstring, which corresponds to a summary with the specified length.

In Fig. 6, we present the quality of summaries generated by the quantum algorithms discussed in the main text, in terms of the three ROUGE metrics. For perspective, we also present the metrics when evaluated against an uniform distribution, and also the metrics for the optimal extractive summary for each article. Note that the ROUGE metrics are obtained by comparing predicted summaries against a human-written "highlight" that comes associated with the CNN/DailyMail dataset. The highlights are brief paraphrasings of the article and do not necessarily contain any sentences or verbatim phrases from the article. As a result, even optimal extractive summaries have the median score that is far from the maximal ROUGE score of 1. Furthermore, as can be seen in Fig. 6, the optimal summaries are only slightly better than the summaries generated by sampling from a uniform distribution and the solutions obtained from the optimization algorithms discussed in the main text only perform as well as random summaries. The fact that even noiseless simulations of quantum algorithms fail to consistently outperform random summaries suggests that the usefulness of the optimization framework as a route to summary generation remains inconclusive, at least for the dataset, evaluation criteria and the qubit counts we consider in our work.

Appendix 5: Tuning hyperparameter λ

The complexity of the optimization of (5) comes from the quadratic terms, which penalize redundancies that are present in the summary. The hyperparameter λ controls the strength of the penalty for including redundant sentences. If the value of λ is very small, the summary will be informative but potentially redundant. Similarly, if λ is too large, the algorithm can pick distinct sentences but with very low information content.

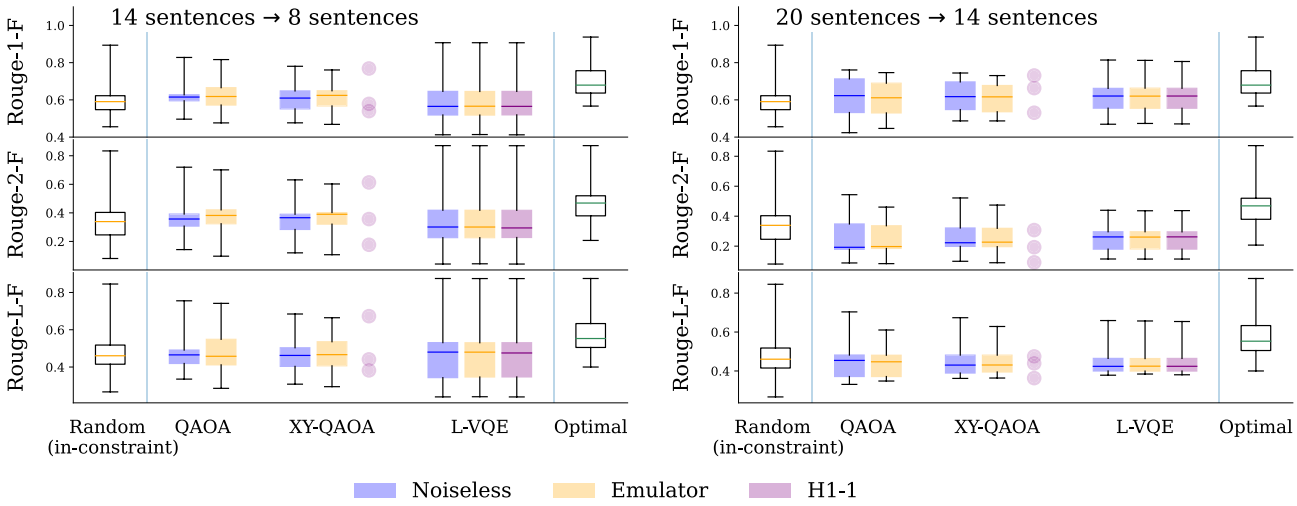

Figure 6. Rouge metrics (1-F, 2-F, and L-F) for extractive summaries generated using the three different optimization algorithms executed in a noiseless simulator, emulator and the real quantum hardware. As helpful guides, we also present the expected ROUGE metric for summaries sampled from the uniform distribution, and also the ROUGE metric for the optimal summary set.

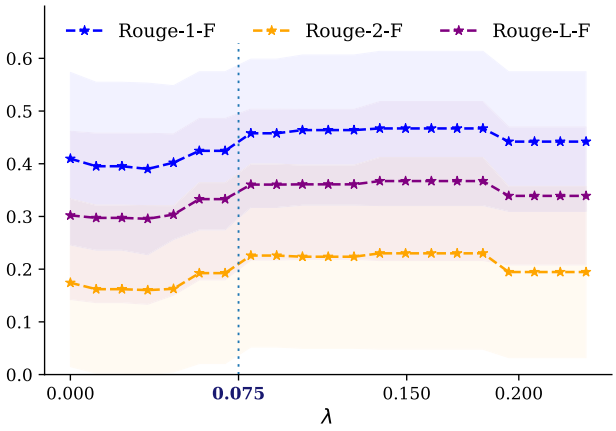

Figure 7. ROUGE metrics for summaries obtained with optimal solutions (obtained via brute-force) for the instances of 14-sentence articles used in the main text, as a function of λ. The dashed vertical line represents the λ used in experiments discussed in the main text. The shaded area represents the standard deviation around the mean.

We study how the penalization for redundancy presented in the summary affects its quality. Specifically, we consider different values of λ and compute the average ROUGE metric, for the 10 optimization instances of 14-sentence articles discussed in the main text. For each article, we generate a set of objective functions 6 for $\lambda \in [0, 0.25]$ and find optimal solutions to these problems with brute-force. We plot the ROUGE metrics for the resulting summaries from the optimal solutions in Fig. 7. We observe a slight increase in average ROUGE for $\lambda > 0.05$. In the experiments discussed in the main text, we use $\lambda = 0.075$.

Received: 12 July 2022; Accepted: 20 September 2022
Published online: 13 October 2022

References

1. Arute, F. *et al.* Quantum supremacy using a programmable superconducting processor. *Nature* **574**, 505–510. https://doi.org/10.1038/s41586-019-1666-5 (2019).
2. Wu, Y. *et al.* Strong quantum computational advantage using a superconducting quantum processor. *Phys. Rev. Lett.*https://doi.org/10.1103/physrevlett.127.180501 *(2021).*
3. Madsen, L. S. *et al.* Quantum computational advantage with a programmable photonic processor. *Nature* **606**, 75–81. https://doi.org/10.1038/s41586-022-04725-x (2022).
4. Sbihi, A. & Eglese, R. W. Combinatorial optimization and green logistics. *4OR* **5**, 99–116. https://doi.org/10.1007/s10288-007-0047-3 (2007).

5. Eskandarpour, M., Dejax, P., Miemczyk, J. & Péton, O. Sustainable supply chain network design: An optimization-oriented review. *Omega* **54**, 11–32. https://doi.org/10.1016/j.omega.2015.01.006 (2015).
6. Kennedy, J. P. *et al.* Application of combinatorial chemistry science on modern drug discovery. *J. Comb. Chem.* **10**, 345–354. https://doi.org/10.1021/cc700187t (2008).
7. Soler-Dominguez, A., Juan, A. A. & Kizys, R. A survey on financial applications of metaheuristics. *ACM Comput. Surv.*https://doi.org/10.1145/3054133 *(2017).*
8. Wang, Z., Hadfield, S., Jiang, Z. & Rieffel, E. G. Quantum approximate optimization algorithm for MaxCut: A fermionic view. *Phys. Rev. A* **97**, 022304. https://doi.org/10.1103/PhysRevA.97.022304 (2018).
9. Zhou, L., Wang, S.-T., Choi, S., Pichler, H. & Lukin, M. D. Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. *Phys. Rev. X* **10**, 021067. https://doi.org/10.1103/PhysRevX.10.021067 (2020).
10. Crooks, G. E. Performance of the quantum approximate optimization algorithm on the maximum cut problem. https://doi.org/10.48550/ARXIV.1811.08419 (2018).
11. Shaydulin, R., Marwaha, K., Wurtz, J. & Lotshaw, P. C. QAOAKit: A toolkit for reproducible study, application, and verification of the QAOA. In *2021 IEEE/ACM Second International Workshop on Quantum Computing Software (QCS)*. https://doi.org/10.1109/qcs54837.2021.00011 (IEEE, 2021).
12. Wurtz, J. & Love, P. MaxCut quantum approximate optimization algorithm performance guarantees for $p > 1$. *Phys. Rev. A* **103**, 042612. https://doi.org/10.1103/PhysRevA.103.042612 (2021).
13. Harrigan, M. P. *et al.* Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. *Nat. Phys* **17**, 332–336 (2021).
14. Shaydulin, R. & Galda, A. Error mitigation for deep quantum optimization circuits by leveraging problem symmetries. In *IEEE International Conference on Quantum Computing and Engineering*, 291–300. https://doi.org/10.1109/QCE52317.2021.00046 (2021).
15. Ebadi, S. *et al.* Quantum optimization of maximum independent set using rydberg atom arrays. *Science* **376**, 1209–1215. https://doi.org/10.1126/science.abo6587 (2022).
16. Bravyi, S., Kliesch, A., Koenig, R. & Tang, E. Hybrid quantum-classical algorithms for approximate graph coloring. *Quantum* **6**, 678. https://doi.org/10.22331/q-2022-03-30-678 (2022).
17. Filippova, K., Surdeanu, M., Ciaramita, M. & Zaragoza, H. Company-oriented extractive summarization of financial news. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, 246–254. (Association for Computational Linguistics, Athens, Greece, 2009) https://doi.org/10.5555/1609067.1609094.
18. Bhattacharya, P., Poddar, S., Rudra, K., Ghosh, K. & Ghosh, S. Incorporating domain knowledge for extractive summarization of legal case documents. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, 22–31 (Association for Computing Machinery, New York, NY, USA, 2021).
19. McDonald, R. A study of global inference algorithms in multi-document summarization. In *Advances in Information Retrieval, ECIR'07* 557–564 (Springer, Berlin, 2007). https://doi.org/10.5555/1763653.1763720.
20. Shaydulin, R., Hadfield, S., Hogg, T. & Safro, I. Classical symmetries and the quantum approximate optimization algorithm. *Quantum Inf. Process.*https://doi.org/10.1007/s11128-021-03298-4 *(2021).*
21. Shaydulin, R., Lotshaw, P. C., Larson, J., Ostrowski, J. & Humble, T. S. Parameter transfer for quantum approximate optimization of weighted MaxCut. https://doi.org/10.48550/ARXIV.2201.11785 (2022).
22. Hadfield, S. *et al.* From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms* **12**, 34. https://doi.org/10.3390/a12020034 (2019).
23. Wang, Z., Rubin, N. C., Dominy, J. M. & Rieffel, E. G. XY-mixers: Analytical and numerical results for the quantum alternating operator ansatz. *Phys. Rev. A.*https://doi.org/10.1103/physreva.101.012320 *(2020).*
24. Liu, X. *et al.* Layer VQE: A variational approach for combinatorial optimization on noisy quantum computers. *IEEE Trans. Quantum Eng.* **3**, 1–20. https://doi.org/10.1109/tqe.2021.3140190 (2022).
25. Farhi, E., Goldstone, J. & Gutmann, S. A quantum approximate optimization algorithm. https://doi.org/10.48550/ARXIV.1411.4028(2014).
26. Hogg, T. & Portnov, D. Quantum optimization. *Inf. Sci.* **128**, 181–197. https://doi.org/10.1016/s0020-0255(00)00052-9 (2000).
27. Shaydulin, R., Safro, I. & Larson, J. Multistart methods for quantum approximate optimization. In *IEEE High Performance Extreme Computing Conference*, 1–8. https://doi.org/10.1109/hpec.2019.8916288 (2019).
28. Wurtz, J. & Lykov, D. Fixed-angle conjectures for the quantum approximate optimization algorithm on regular MaxCut graphs. *Phys. Rev. A* **104**, 052419. https://doi.org/10.1103/PhysRevA.104.052419 (2021).
29. Sherrington, D. & Kirkpatrick, S. Solvable model of a spin-glass. *Phys. Rev. Lett.* **35**, 1792–1796. https://doi.org/10.1103/PhysRevLett.35.1792 (1975).
30. Farhi, E., Goldstone, J., Gutmann, S. & Zhou, L. The quantum approximate optimization algorithm and the Sherrington-Kirkpatrick model at infinite size. https://doi.org/10.48550/ARXIV.1910.08187 (2019).
31. Basso, J., Farhi, E., Marwaha, K., Villalonga, B. & Zhou, L. The quantum approximate optimization algorithm at high depth for MaxCut on large-girth regular graphs and the Sherrington–Kirkpatrick model. https://doi.org/10.48550/arXiv.2110.14206 (2021).
32. Xu, J., Gan, Z., Cheng, Y. & Liu, J. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5021–5031 (Association for Computational Linguistics, Online, 2020) https://doi.org/10.18653/v1/2020.acl-main.451.
33. Zhong, M. *et al.* Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6197–6208 (Association for Computational Linguistics, Online, 2020) https://doi.org/10.18653/v1/2020.acl-main.552.
34. Liu, Y. Fine-tune BERT for extractive summarization. https://doi.org/10.48550/ARXIV.1903.10318 (2019).
35. Document summarization on cnn/daily mail. https://paperswithcode.com/sota/document-summarization-on-cnn-daily-mail. Accessed 6 Oct 2022.
36. Hermann, K. M. *et al.* Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems—Volume 1*, NIPS'15, 1693–1701 (MIT Press, Cambridge, MA, USA, 2015).
37. Reimers, N. & Gurevych, I. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3982–3992 (Association for Computational Linguistics, Hong Kong, China, 2019) https://doi.org/10.18653/v1/D19-1410.
38. Aizawa, A. An information-theoretic perspective of tf-idf measures. *Inf. Process. Manage* **39**, 45–65. https://doi.org/10.1016/S0306-4573(02)00021-3 (2003).
39. Aleksandrowicz, G. *et al.* Qiskit: An open-source framework for quantum computing. https://doi.org/10.5281/zenodo.2562111 (2019).
40. Sivarajah, S. *et al.* t|ket>: A retargetable compiler for NISQ devices. *Quantum Sci. Technol.* **6**, 014003. https://doi.org/10.1088/2058-9565/ab8e92 (2020).
41. Bärtschi, A. & Eidenbenz, S. Deterministic preparation of Dicke states. In *Fundamentals of Computation Theory* 126–139 (Springer International Publishing, 2019). https://doi.org/10.1007/978-3-030-25027-0_9.
42. Mukherjee, C. S., Maitra, S., Gaurav, V. & Roy, D. On actual preparation of Dicke state on a quantum computer. https://doi.org/10.48550/ARXIV.2007.01681 (2020).

43. Aktar, S., Bärtschi, A., Badawy, A.-H.A. & Eidenbenz, S. A divide-and-conquer approach to Dicke state preparation. *IEEE Trans. Quantum Eng.* **3**, 1–16. https://doi.org/10.1109/TQE.2022.3174547 (2022).
44. McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R. & Neven, H. Barren plateaus in quantum neural network training landscapes. *Nat. Commun.*https://doi.org/10.1038/s41467-018-07090-4 *(2018).*
45. Holmes, Z., Sharma, K., Cerezo, M. & Coles, P. J. Connecting ansatz expressibility to gradient magnitudes and barren plateaus. *PRX Quantum.*https://doi.org/10.1103/prxquantum.3.010313 *(2022).*
46. Pino, J. M. *et al.* Demonstration of the trapped-ion quantum CCD computer architecture. *Nature* **592**, 209–213. https://doi.org/10.1038/s41586-021-03318-4 (2021).
47. Ryan-Anderson, C. *et al.* Realization of real-time fault-tolerant quantum error correction. *Phys. Rev. X* **11**, 041058. https://doi.org/10.1103/PhysRevX.11.041058 (2021).
48. Otterbach, J. S. *et al.* Unsupervised machine learning on a hybrid quantum computer. https://doi.org/10.48550/ARXIV.1712.05771 (2017).
49. Lacroix, N. *et al.* Improving the performance of deep quantum optimization algorithms with continuous gate sets. *PRX Quantum* **1**, 110304. https://doi.org/10.1103/PRXQuantum.1.020304 (2020).
50. Baker, J. S. & Radha, S. K. Wasserstein solution quality and the quantum approximate optimization algorithm: A portfolio optimization case study. https://doi.org/10.48550/ARXIV.2202.06782 (2022).
51. Qiang, X. *et al.* Large-scale silicon quantum photonics implementing arbitrary two-qubit processing. *Nat. Photon.* **12**, 534–539. https://doi.org/10.1038/s41566-018-0236-y (2018).
52. Willsch, M., Willsch, D., Jin, F., Raedt, H. D. & Michielsen, K. Benchmarking the quantum approximate optimization algorithm. *Quantum Inf. Process.*https://doi.org/10.1007/s11128-020-02692-8 *(2020).*
53. Abrams, D. M., Didier, N., Johnson, B. R., da Silva, M. P. & Ryan, C. A. Implementation of XY entangling gates with a single calibrated pulse. *Nat. Electron.* **3**, 744–750. https://doi.org/10.1038/s41928-020-00498-1 (2020).
54. Bengtsson, A. *et al.* Improved success probability with greater circuit depth for the quantum approximate optimization algorithm. *Phys. Rev. Appl.*https://doi.org/10.1103/physrevapplied.14.034010 *(2020).*
55. Earnest, N., Tornow, C. & Egger, D. J. Pulse-efficient circuit transpilation for quantum applications on cross-resonance-based hardware. *Phys. Rev. Res.*https://doi.org/10.1103/physrevresearch.3.043088 *(2021).*
56. Santra, G. C., Jendrzejewski, F., Hauke, P. & Egger, D. J. Squeezing and quantum approximate optimization. https://doi.org/10.48550/ARXIV.2205.10383 (2022).
57. Kakkar, A., Larson, J., Galda, A. & Shaydulin, R. Characterizing error mitigation by symmetry verification in QAOA. arXiv:2204.05852 (2022).
58. Shaydulin, R., Ushijima-Mwesigwa, H., Safro, I., Mniszewski, S. & Alexeev, Y. Network community detection on small quantum computers. *Adv. Quantum Technol.* **2**, 1900029. https://doi.org/10.1002/qute.201900029 (2019).
59. Ushijima-Mwesigwa, H. *et al.* Multilevel combinatorial optimization across quantum architectures. *ACM Trans. Quantum Comput.* **2**, 1–29. https://doi.org/10.1145/3425607 (2021).
60. Golden, J., Bärtschi, A., O'Malley, D. & Eidenbenz, S. QAOA-based fair sampling on NISQ devices. https://doi.org/10.48550/ARXIV.2101.03258 (2021).
61. Bartschi, A. & Eidenbenz, S. Grover mixers for QAOA: Shifting complexity from mixer design to state preparation. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, 2020) https://doi.org/10.1109/qce49297.2020.00020.
62. Pelofske, E., Golden, J., Bärtschi, A., O'Malley, D. & Eidenbenz, S. Sampling on nisq devices: "who's the fairest one of all?". In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 207–217. https://doi.org/10.1109/QCE52317.2021.00038 (2021).
63. Fingerhuth, M., Babej, T. & Ing, C. A quantum alternating operator ansatz with hard and soft constraints for lattice protein folding. https://doi.org/10.48550/ARXIV.1810.13411 (2018).
64. Pagano, G. *et al.* Quantum approximate optimization of the long-range ising model with a trapped-ion quantum simulator. *Proc. Natl. Acad. Sci* **117**, 25396–25401. https://doi.org/10.1073/pnas.2006373117 (2020).
65. Herman, D. *et al.* A survey of quantum computing for finance. https://doi.org/10.48550/ARXIV.2201.02773 (2022).
66. Pistoia, M. *et al.* Quantum machine learning for finance iccad special session paper. In *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, 1–9. https://doi.org/10.1109/ICCAD51958.2021.9643469 (2021).
67. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186 (Association for Computational Linguistics, Minneapolis, Minnesota, 2019) https://doi.org/10.18653/v1/N19-1423.
68. Vaswani, A. *et al.* Attention is all you need. In *Advances in Neural Information Processing Systems* Vol. 30 (eds Guyon, I. *et al.*) (Curran Associates, Inc., 2017).
69. Wang, C., Li, M. & Smola, A. J. Language models with transformers. https://doi.org/10.48550/ARXIV.1904.09408 (2019).
70. Achananuparp, P., Hu, X. & Shen, X. The evaluation of sentence similarity measures. In *Data Warehousing and Knowledge Discovery* (eds Song, I.-Y. *et al.*) 305–316 (Springer, 2008). https://doi.org/10.1007/978-3-540-85836-2_29.
71. Zheng, H. & Lapata, M. Sentence centrality revisited for unsupervised summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 6236–6247 (Association for Computational Linguistics, Florence, Italy, 2019) https://doi.org/10.18653/v1/P19-1628.
72. Powell, M. J. D. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in Optimization and Numerical Analysis* (eds Gomez, S. & Hennart, J.-P.) 51–67 (Springer, 1994). https://doi.org/10.1007/978-94-015-8330-5_4.
73. Powell, M. Direct search algorithms for optimization calculations. *Acta Numer.* **7**, 287–336. https://doi.org/10.1017/S0962492900002841 (1998).
74. Quantinuum system model H1 product data sheet [retrieved 08/30/2022]. https://www.quantinuum.com/products/h1.
75. Lin, C.-Y. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out* 74–81 (Association for Computational Linguistics, 2004).

Acknowledgements

The authors thank Tony Uttley, Brian Neyenhuis, Jenni Strabley and the whole Quantinuum team for their support and feedback, and especially for providing us preview access to the Quantinuum H1-1 upgraded to 20 qubits. The authors thank Andreas Bärtschi and Stephan Eidenbenz for the helpful discussions on the Dicke-state preparation. Additionally, the authors appreciate the support of their FLARE colleagues at JPMorgan Chase. P.N. acknowledges funding by the DoE ASCR Accelerated Research in Quantum Computing program (award No. DE-SC0020312), DoE QSA, NSL QLCI (award No. OMA-2120757), NSF PFCQC program, DoE ASCR Quantum Testbed Pathfinder program (award No. DE-SC0019040), U.S. Department of Energy Award No. DE-SC0019499, AFOSR, ARO MURI, AFOSR MURI, and DARPA SAVaNT ADVENT.

Disclaimer

This paper was prepared for information purposes by the Future Lab for Applied Research and Engineering (FLARE) group of JPMorgan Chase Bank, N.A.. This paper is not a product of the Research Department of JPMorgan Chase Bank, N.A. or its affiliates. Neither JPMorgan Chase Bank, N.A. nor any of its affiliates make any explicit or implied representation or warranty and none of them accept any liability in connection with this paper, including, but limited to, the completeness, accuracy, reliability of 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 a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction.

Author contributions

P.N., R.Y. devised and implemented the optimization formulation. R.S., R.Y. implemented the quantum algorithms. P.N., R.S., R.Y., D.H. performed the experiments. P.N., R.S., R.Y. analyzed the data. M.P. supervised the project. All authors contributed to the technical discussions, evaluation of the results, and the writing of the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to R.S.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.