RESEARCH   **Open Access**

Exponential qubit reduction in optimization for financial transaction settlement

Elias X. Huber[1,2*], Benjamin Y.L. Tan[2], Paul R. Griffin[4] and Dimitris G. Angelakis[2,3,5]

*Correspondence:
elixaver@gmail.com
[1]D-CHAB Interdisciplinary Sciences,
ETH Zürich, Vladimir-Prelog-Weg
1-5/10, 8093, Zurich, Switzerland
[2]Centre for Quantum Technologies,
National University of Singapore, 3
Science Drive 2, Singapore, 117543,
Singapore
Full list of author information is
available at the end of the article

Abstract

We extend the qubit-efficient encoding presented in (Tan et al. in Quantum 5:454, 2021) and apply it to instances of the financial transaction settlement problem constructed from data provided by a regulated financial exchange. Our methods are directly applicable to any QUBO problem with linear inequality constraints. Our extension of previously proposed methods consists of a simplification in varying the number of qubits used to encode correlations as well as a new class of variational circuits which incorporate symmetries thereby reducing sampling overhead, improving numerical stability and recovering the expression of the cost objective as a Hermitian observable. We also propose optimality-preserving methods to reduce variance in real-world data and substitute continuous slack variables. We benchmark our methods against standard QAOA for problems consisting of 16 transactions and obtain competitive results. Our newly proposed variational ansatz performs best overall. We demonstrate tackling problems with 128 transactions on real quantum hardware, exceeding previous results bounded by NISQ hardware by almost two orders of magnitude.

Keywords: Quantum Computing; Quantum Optimization; NISQ; QUBO; Mixed binary optimization; Quantum Finance; Qubit reduction

1 Introduction

Provable asymptotic advantages of quantum computing over classical algorithms have been shown in the *fault-tolerant* regime ([1, 2]) and *quantum computational supremacy* ([3]) has been claimed experimentally in circuit sampling tasks ([4–6]).[1] Methods that promise to extend these computational advantages to *relevant* problems with *available noisy intermediate scale quantum* (NISQ) devices have been an active field of research over the past decade. A recent breakthrough in this regard was achieved by IBM Quantum ([8]), claiming evidence for the utility of said NISQ devices by simulating the evolution under an Ising Hamiltonian beyond the reach of standard[2] classical simulation methods. Most research to this end of useful NISQ algorithms is concerned with problems in Hamil-

---

[1]Although some problem instances have later been shown to be classically simulable, e.g. [7].

[2]Efficient classical simulation of the experiment was claimed shortly thereafter, e.g. using tensor networks: [9–11].

Springer

tonian simulation, machine learning or energy minimization/optimization ([12, 13]). This work concerns the latter.

Outline of this paper　　In the introduction, we give an overview of quantum optimization in NISQ, summarize different approaches to reduce the number of qubits (1.1) and introduce the transaction settlement problem (1.2). We extend the qubit reduction technique introduced in [14] to find approximate solutions to problem instances larger than previously attempted. We outline the mapping used between the quantum state and the binary variables of the problem (2.2) and how the cost can be estimated using this quantum state (2.3). We introduce a new variational ansatz derived to incorporate symmetries of the encoding scheme (2.4), before concluding with simulation (3.1) and quantum hardware (3.2) results.

1.1 Quantum optimization – quadratic unconstrained binary optimization

The optimization problem we consider in 1.2 will generalize *quadratic unconstrained binary optimization* (QUBO) problems, which have the form

$$\arg\min_{\mathbf{x}\in\{0,1\}^I} C(\underline{\mathbf{x}}) = \arg\min_{\mathbf{x}\in\{0,1\}^I} \underline{\mathbf{x}}^T Q \underline{\mathbf{x}} \tag{1}$$

where I is the number of binary entries of the vector $\underline{\mathbf{x}}$ and Q is any real (usually symmetric) matrix, $Q \in \mathbb{R}^{I \times I}$. Finding the vector $\underline{\mathbf{x}}$ minimizing equ. (1) for general Q is NP hard ([15]). Many combinatorial/graph problems such as MaxCut can be readily mapped to QUBO problems and a wide range of industrial applications is known. This includes training of machine learning models ([16]) and optimization tasks such as assignment problems ([17]), route optimization ([18]) or - the focus of this study - financial transaction settlement ([19]).[3] This broad applicability and (by benchmarking existing classical solvers) "verifiable" advantage make QUBO problems a great test-bed in the search for a useful quantum advantage.

The solution of equation (1) corresponds to the ground state of an Ising Hamiltonian H_Q on I qubits,

$$H_Q = \frac{1}{4}\sum_{i,j=1}^{I} Q_{ij}(1 - \sigma_z^i)(1 - \sigma_z^j), \tag{2}$$

with σ_a^i referring to the Pauli operator a on qubit i. This allows mapping a QUBO problem on I variables to the problem of finding the ground state of a Hamiltonian on I qubits. We extend equation (2) in Sect. 2.2 and 2.3 by applying the qubit compression from [14] to reduce the number of qubits to $O(\log I)$ at the cost of losing the formulation (2) as the ground state of a Hermitian operator. *Quantum solvers* (QS) to the Ising Hamiltonian or more general ground-state problems have been studied extensively. A short overview is given in Table 1 and the following:

---

[3]For further applications see Chap. 2 in [20].

Annealing   Introduced as early as 1994 ([21]) and inspired by simulated annealing ([22]), *quantum annealing* aims to find the ground state of H_Q in (2) by *adiabatically* transforming $H_{\text{tot}}(t) = s(t)H_Q + (1 - s(t))H_m$, with the *mixing Hamiltonian* $H_m = \sum_{j=1}^{I} \sigma_j^x$, over a time span $t \in [0, t_{\text{end}}]$. Here, $s(t)$ is the annealing schedule, with $s(0) = 0$ and $s(t_{\text{end}}) = 1$. Reading out the state of the annealing device at the end of this transformation yields candidates for the optimal solution $\underline{\mathbf{x}}$. Annealing devices are not guaranteed to find optimal solutions efficiently and can only implement a limited set of Hamiltonians, often restricted in their connectivity (resulting in limitations on the non-zero entries of Q) ([23]). Despite these limitations, general-purpose QUBO solvers based on hybrid classical computation and quantum annealing are commercially available with as many as 5000 (1 million) physical nodes (variables, I in equ. (1)) for D-wave's Advantage$^{\text{TM}}$ annealer ([24]).[4]

QAOA   **Q**uantum **A**pproximate **O**ptimization **A**lgorithms ([25]) can be regarded as implementing a parametrized, *trotterized* version of the quantum annealing schedule on gate-model based quantum computers. The parameterized p-layered circuit $e^{-iH_Q\beta_p}e^{-iH_m\gamma_p}...e^{-iH_Q\beta_1}e^{-iH_m\gamma_1}$ is applied to $|+\rangle^{\otimes I}$ and measured in the computational basis. This yields candidate vectors $\underline{\mathbf{x}}$ by identifying each binary variable with one qubit. The parameters $\{\beta_j, \gamma_j\}$ are classically optimized to minimize equ. (1) (minimize $\langle H_Q \rangle$). QAOA provides theoretical guarantees in its convergence to the exact solution for $p \to \infty$ given optimal parameters. Yet, implementing the evolution of H_Q and reaching sufficient depth p on NISQ devices can be infeasible in the case of many non-zero entries of Q.

Hardware-efficient VQA   In this work, we make use of general *Variational Quantum Algorithms* to minimize a cost estimator (in the context of quantum chemistry often referred to as VQE, *variational quantum eigensolver* ([26]), and applied beyond Ising Hamiltonians). VQAs are general quantum circuit ansätze parameterized by classical parameters, hence QAOA can be seen as a special case of a VQA. We use the term *hardware-efficient* VQA loosely for ansätze whose gates, number of qubits and circuit depth suit current NISQ devices. Analogously to QAOA, the parameters of the VQA circuit are optimized classically through evaluation of some classical cost function on the measured bit-vector. As we will see later, this cost function does not necessarily correspond to a Hermitian observable. VQAs are widely studied in the NISQ era beyond their application to combinatorial optimization problems ([27–29]). Challenges, most notably vanishing gradients for expressive circuits ([30–32]) and remedies ([33–41]) exist aplenty but will not play a central role in this paper. While the generality of VQAs allows for tailored hardware-efficient ansätze which are independent of the problem itself, this comes at the cost of losing the remaining theoretical guarantees of QAOA and adiabatic ground state computation.

Non-VQA, quantum-assisted solvers   Other quantum algorithms for solving ground-state problems have been proposed in the literature. Examples include quantum-assisted algorithms, often inspired by methods such as Krylov subspace, imaginary time evolution or quantum phase estimation. For example, quantum computers are used to calculate overlaps between quantum states employed in a classical outer optimization loop ([42–48]). Although some of these approaches are variational in the circuit ansatz, they

---

[4]Note that it is not public how exactly the quantum annealer is used as a subroutine in this hybrid computation.

do not directly correspond to the classical-quantum feedback loop in the VQA setting described above and are beyond the focus of this work.

Classical solvers   It should be noted at this point, that approaches using classical computing for tackling QUBO problems exist. Among them[5] are general purpose optimization suites such as Gurobi ([49]), CPLEX ([50]) or SCIP ([51]) as well as dedicated approximation algorithms such as simulated annealing ([52]), TABU search ([53])) or the relaxation-based Goemans and Williamson ([54]) algorithm which guarantees an approximation ratio of at least 0.878[6] for Max-Cut problems. Due to the NP-hardness of the general problem, all classical solvers are either approximations or have no polynomial worst-case runtime guarantees.

NISQ-limitations   Quantum computers are not expected to break NP-hardness (cf. [56, 57] and the lack of any polynomial-time quantum algorithm for an NP-hard problem) and it is often justified to regard quantum approaches to QUBO as heuristics *hoped* to provide practical advantages rather than general purpose solvers with rigorous runtime *and* optimality guarantees. This makes benchmarking on relevant problem instances paramount in guiding the search for promising quantum algorithms. Yet, most NISQ-era quantum approaches suffer from a combination of

1. **Problem size** limited by the number of available qubits
2. Constraints on the **problem class** (connectivity of Q)

making a direct application of QS to *relevant* problem instances infeasible on NISQ-devices ([58]). While 1. is a consequence of the limited number of qubits available on NISQ devices, 2. can be seen as a consequence of noise in the qubit and operations: Computations become infeasible due to low coherence times and noisy gates paired with often deep circuits (e.g. arising from the limited lattice-connectivity of devices based on superconducting qubits) upon decomposition into hardware-native gates. Constraints on the problem class can also arise from the fundamental design of the algorithm itself.

How these limitations on problem size and class apply to the different QS is summarized in Table 1. Various work has been done to address these challenges. Improved problem embeddings ([59]), decomposition ([60]), compilation and hardware-efficient ansätze are just some approaches to deal with connectivity issues. A wide variety of qubit-reduction methods has been suggested in the quantum optimization and quantum chemistry literature, see Table 2.

Proposing a solution to the limitations in Table 1 and pushing the boundaries of QUBO problems accessible by QS is a central motivation for this work. We give a detailed description of our qubit-reduction method in Sect. 2.2.

1.2 Financial transaction settlement

We refer to the *transaction settlement problem* as a computational task, consisting of parties $\{1, \ldots, K\}$ with balances $\{\underline{\mathbf{bal}}_k\}$ submitting trades $\{1, \ldots, I\}$ to a clearing house. The task faced by the clearing house is to determine the maximal set of transactions that can be executed without any party k falling below its credit limit $\underline{\mathbf{lim}}_k$. An overview of the notation

---

[5]A more extensive overview can be found in Chap. 11 of [20].

[6]Which is optimal for any polynomial-time classical algorithm assuming the unique games conjecture ([55]).

Table 1 Overview of QS for ground state problems and NISQ limitations in the "vanilla" formulations of these approaches. Shown is the relation between physical qubits and the number of variables as well as the impact of the problem connectivity. Grey underlaid will be the focus in this work: Our qubit-efficient encoding makes use of VQA and we will benchmark our results against QAOA

	Algorithm			
	Quantum solvers/heuristics (QS)			
NISQ limitation	Annealing	QAOA	(Hardware-efficient) VQA / VQE	Quantum assisted solver
#Variables l	#qubits = l; realized experimentally: $l \leq 5000$	#qubits = l; realized experimentally: $l \leq 127$		#qubits = l
Connectivity: $Q_{ij} \overset{!}{=} 0$ for some $i \neq j$	restricted to device connectivity	problem vs. device connectivity $\leftrightarrow$ circuit depth	ansatz-dependent	ansatz-dependent, overlap calculation
References	[24]	[61–65]	[19, 26, 27, 29]	[42–48]

Figure 1 Example of a transaction settlement problem constructed using data provided from a regulated financial exchange. Eight transactions (arrows) between seven parties (numbered squares) are depicted. Each party has initial balances for cash ($) and different securities (S1, …). The optimal solution which settles the maximal amount of transactions without violating balance constraints is indicated through ✓ and ✗. The solution is not unique, another optimal solution would settle T2 instead of T3. Even for a problem of only eight transactions, non-trivial dependencies between different transactions exist: For example, T4 can only be settled if T5 is settled which in turn requires T8

is given in Table 3 and a graph representation of a transaction settlement problem with parties as nodes and transactions as edges is shown in Fig. 1.

In the case when not all parties have sufficient balances to meet all settlement instructions they are involved with, finding this maximal set can be difficult with classical computing resources. Intuitively, this is because a party's ability to serve outgoing transactions may depend on its incoming transactions, creating many interdependencies between different parties (cf. Fig. 1). Whilst classical technology is sufficient for current transaction volumes, increases could be expected from more securities in emerging markets and digital tokens, for example. Furthermore, cash shortages make optimization more challenging as it becomes harder to allocate funds optimally among various settlement obligations, determining the priority of different trades and parties and an increased risk of settlement failures. Quantum technologies offer a potential path to mitigate these issues.

Transactions can be conducted both in currencies and securities such as equity and bonds (hence $\underline{\mathbf{bal}}_k$ and $\underline{\mathbf{lim}}_k$ are vector-valued). A financial exchange may, for example, handle as many as one million trades involving 500-600 different securities by up to 100 financial institutions (parties) per day.

Table 2 Overview of different methods proposed in the literature to reduce the number of qubits needed in ground state problems. Not all methods are directly applicable to arbitrary QUBO Hamiltonians. Different columns correspond to a rough qualitative classification by the author, grey underlaid what this work is based on. Quantum solvers in row "QS applicable" are set in braces if they can only be used with some of the references in that column or if further restrictions apply. In the row "Caveats" we summarize limitations or additional overhead incurred by the methods: evaluation of more circuits, more shots or Pauli measurements, more connectivity-demanding circuits (for shallow decompositions) or deeper circuits. This table focuses on methods that can be used to reduce the number of qubits, applicable in the framework of at least one of the QS in table 1. Qubit-efficient approaches to QUBO problems exist beyond that, e.g. [66]

Features	Reduction method					
	Problem reduction	Divide & Conquer		Binary encoding qubit compression	Qubit efficient relaxation	Qubit-reuse
		Objective	Quantum circuit			
Description	Pre-processing to simplify problem	Split in smaller problems and recombine		Replace one-hot enc. by binary enc.	Map to relaxation	Mid-circuit msm. & reset
QS applicable	All	All	All gate-based	VQA (QAOA)	(VQA)	VQA
Qubit reduction	problem-dependent	$\times\frac{1}{c}$ or set to constant		exponential	$\times\frac{1}{c}$ / exponential	set to constant
Caveats	No guaranteed reduction	Divide/Conquer impacted by connectivity, circuit overhead		msm. overhead	$c \leftrightarrow$ problem connectivity & msm. overhead	Depth & msm. overhead
References	[67, 68]	[61–65, 69]	circ. cutting: [70]; clustering: [71]; causal cones: [72]	here: [14]; other: [73–75]	[76, 77] / [78, 79]	[80]

Table 3 The left part of this table shows the format of the settlement instructions data samples obtained from a regulated financial exchange. Each settlement instruction consists of a sending (PARTICIPANT) and receiving (COUNTERPARTY) party, specifies the security transacted (SECURITY), the quantity traded (QUANTITY) as well as the countervalue (CONSIDERATION) in Singapore dollars. For SETTLEMENT_TYPE *Delivery Vs Payment* (DVP), a security is traded against a cash settlement. Here, the only alternative is *Free Of Payment* (FOP), in which case only the security is transferred from the seller to the buyer. The right part shows the corresponding problem inputs and their mathematical notation (adapted from [19])

Financial exchange data	Example	Inputs	Math notation
SETTLEMENT_INSTRUCTION		transactions	$i \in \{1,\ldots,I\}$
PARTICIPANT	205	parties	$k \in \{1,\ldots,K\}$
COUNTERPARTY	270		
INSTRUMENT	nc157	currencies/securities	$j \in \{1,\ldots,J\}$
QUANTITY	1300	transaction value (security)	$\underline{\mathbf{v}}_{ik} \in \mathbb{R}^J$
CONSIDERATION	441.85	transaction value (currency)	
transaction weights set to one		transaction weights	$\underline{\mathbf{w}} \in \mathbb{R}^I_{\geq 0}$
credit limit absorbed in balance		credit limits	$\underline{\mathbf{lim}}_k \in \mathbb{R}^J$
balance generated		balance	$\underline{\mathbf{bal}}_k \in \mathbb{R}^J$
SETTLEMENT_TYPE	DVP		
		Decision Variables	
		settle transaction or not	$\underline{\mathbf{x}} \in \{0,1\}^I$
		slack variables	$\underline{\mathbf{s}}_k \in \mathbb{R}^J_{\geq 0}$

QUBO formulation   To obtain a QUBO formulation of the transaction settlement problem, we follow a slightly simplified version of [19]. The mathematical formulation as a binary optimization problem with inequality constraints looks as follows:

$$\arg\max_{\underline{\mathbf{x}}} \ \underline{\mathbf{w}}^T \underline{\mathbf{x}} \tag{3}$$

$$\text{subject to } \textit{balance constraints}: \sum_i x_i \underline{\mathbf{v}}_{ik} + \underline{\mathbf{bal}}_k - \underline{\mathbf{lim}}_k \geq 0 \ \forall k \in \{1,\ldots,K\} \tag{4}$$

where for generality, a weight w_i is given for each transaction i and $\underline{\mathbf{v}}_{ik}$ represents the balance changes (in cash and securities) for party k in transaction i. In practice one might choose w_i proportional to the transaction value of transaction i, for simplicity, we will always choose $w_i \equiv 1$.

The solution of this linear constrained binary optimization problem equals the solution of the mixed binary optimization (MBO)

$$\arg\max_{\substack{\underline{\mathbf{x}} \\ \underline{\mathbf{s}}_k \geq 0}} \left[\underline{\mathbf{w}}^T \underline{\mathbf{x}} - \lambda \sum_{k=1}^{K} \left(\sum_{i=1}^{I} x_i \underline{\mathbf{v}}_{ik} + \underline{\mathbf{bal}}_k - \underline{\mathbf{lim}}_k - \underline{\mathbf{s}}_k \right)^2 \right] \tag{5}$$

for large λ, referred to as the *slack parameter*. Here, continuous *slack variables* $\underline{\mathbf{s}}_k \geq 0$ (element-wise) were introduced to capture the inequality constraints as penalty terms in the objective. Note, that by approximating $\underline{\mathbf{s}}_k$ as a binary representation, i.e. $(\underline{\mathbf{s}}_k)_i \approx \sum_{l=-L_1}^{l=L_2} \tilde{b}_{kil} 2^l, \tilde{b}_{kil} \in \{0,1\}$, the problem could further be transformed into a QUBO problem without any constraints. For large enough λ, any violation of the constraints (4) will result in a less-than-optimal solution vector. Equation (5) can directly be rewritten:

$$\arg\min_{\substack{\underline{\mathbf{x}} \\ \underline{\mathbf{s}}_k \geq 0}} \underline{\mathbf{x}}^T A \underline{\mathbf{x}} + \underline{\mathbf{b}}(\underline{\mathbf{s}})^T \underline{\mathbf{x}} + c(\underline{\mathbf{s}}) \tag{6}$$

where: $A := -\lambda V V^T$,

with $V \in \mathbb{R}^{I \times KJ}$, $V_{il} := (\underline{\mathbf{v}}_{ik(l)})_{j(l)}$ $\left[\text{with } k(l) := \lceil \frac{l}{J} \rceil, j(l) := (l \bmod J) + 1 \right]$ \hfill (7)

$$b_i(\underline{\mathbf{s}}) := w_i - 2\lambda \sum_{k=1}^{K} \left[\underline{\mathbf{bal}}_k - \underline{\mathbf{lim}}_k - \underline{\mathbf{s}}_k \right] \underline{\mathbf{v}}_{ik} \hfill (8)$$

$$c(\underline{\mathbf{s}}) := -\lambda \sum_{k=1}^{K} \left[\underline{\mathbf{bal}}_k - \underline{\mathbf{lim}}_k - \underline{\mathbf{s}}_k \right]^2 \hfill (9)$$

For fixed $\underline{\mathbf{s}}$, the minimization over binary $\underline{\mathbf{x}}$ is a QUBO problem as in equ. (1) with $Q = A + \text{Diag}[\underline{\mathbf{b}}(\underline{\mathbf{s}})]$,[7] where $\text{Diag}[\underline{\mathbf{b}}]$ is the matrix with the vector $\underline{\mathbf{b}}$ on the diagonal and zeros elsewhere.

1.3 Contribution of this work

The structure of this work is as follows: In Sect. 2.1, we outline the construction of the transaction settlement instances from data provided by a regulated financial exchange. In Sect. 2.2, we use the encoding scheme listed in [14] to reduce the number of qubits required and extend the ideas to include a new variational cost objective and ansatz. Section 3.1 presents the results, using the transaction settlement instances generated as a testbed for comparing our methods with QAOA and exploring different encodings. Section 3.2 offers comparisons in the solutions obtained when using the exponential qubit reduction to tackle problems with 128 transactions on real quantum hardware by IonQ and IBM Quantum, exceeding previous results using quantum hardware [19]. We present some analysis regarding the results obtained, before concluding with Sect. 4. To the best of our knowledge, this is the first work that tackles mixed binary optimization problems with a qubit-efficient approach on a quantum computer.

2 Methodology

To give an overview of the methodology, we first detail how the financial settlement problem can be constructed using data provided by a regulated financial exchange (2.1). We will then describe the quantum algorithm (Fig. 2) consisting of a heuristic to exponentially reduce the number of qubits (Sect. 2.2), a cost-objective (Sect. 2.3), a parameterized quantum circuit to generate solution bit-vectors (Sect. 2.4) and finally the classical optimization (Sect. 2.5).
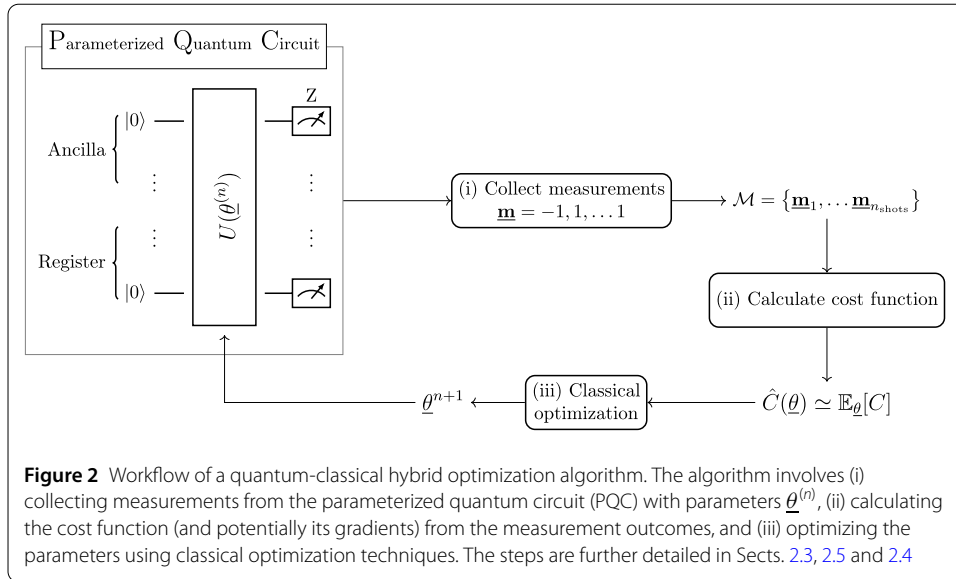
2.1 Problem instance - financial transaction settlement

Dataset   This work uses anonymized transaction data to generate settlement problems of arbitrary size I. The format of the settlement instructions made available for this purpose can be seen in Table 3. To generate a problem instance we proceeded as follows:

1. Fix the number of transactions (I), number of parties (K) and an integer $R \leq I$.
2. Choose $I - R$ random transactions from the dataset and randomly assign them to parties (sender and recipient). A single *transaction* consists of a security being transferred from one party to the other and (if delivery vs payment) a cash transaction in the other direction.[8]

---

[7]Here we use: $\underline{\mathbf{x}}^T \text{Diag}[\underline{\mathbf{b}}]\underline{\mathbf{x}} = \sum_{i=1}^{I} b_i x_i^2 \overset{x_i^2 = x_i}{=} \underline{\mathbf{b}}^T \underline{\mathbf{x}}$.

[8]To avoid confusion, we will refer to *transfers* instead when only considering a single security or cash transfer.

Figure 2 Workflow of a quantum-classical hybrid optimization algorithm. The algorithm involves (i) collecting measurements from the parameterized quantum circuit (PQC) with parameters $\underline{\theta}^{(n)}$, (ii) calculating the cost function (and potentially its gradients) from the measurement outcomes, and (iii) optimizing the parameters using classical optimization techniques. The steps are further detailed in Sects. 2.3, 2.5 and 2.4

3. As our dataset does not provide account balances or credit limits, we set $\underline{\mathbf{lim}}_k = \underline{\mathbf{0}}$ and choose minimal non-negative balances $\underline{\mathbf{bal}}_k$ for each party k such that all previously chosen transactions can be jointly executed without any party's balance becoming negative. The balances hence depend on the first $I - R$ transactions chosen. This choice of $\underline{\mathbf{bal}}_k$ is made by considering the net balance-change for each party if all transactions were conducted.
4. Choose additional R random transactions from the dataset and randomly assign them to parties (without changing the balances assigned in the previous step).

This procedure ensures the optimal solution contains at least $I - R$ valid transactions. Due to the minimal choice of the balances, most of the R transactions chosen last are expected to be invalid in the optimal solution.

To mitigate large differences in transaction volumes between different parties (S$ $10 - 10^6$) as well as different units (cash and different securities) in the data samples, we renormalize each party's balance, credit limit and transaction volume:

$$\forall \text{parties } k \in \{1, \ldots, K\}, \forall \text{securities } j \in \{1, \ldots, J\} : \tag{10}$$

$$\gamma_{kj} := \text{mean}\left(\{|(v_{ik})_j|\}_{i=1}^I \setminus \{0\}\right) \tag{11}$$

$$(v_{ik})_j \mapsto \frac{(v_{ik})_j}{\gamma_{kj}} \; \forall i \in \{1, \ldots, I\} \tag{12}$$

$$(\underline{\mathbf{lim}}_k)_j \mapsto \frac{(\underline{\mathbf{lim}}_k)_j}{\gamma_{kj}} \tag{13}$$

$$(\underline{\mathbf{bal}}_k)_j \mapsto \frac{(\underline{\mathbf{bal}}_k)_j}{\gamma_{kj}} \tag{14}$$

which, using equ (3) and (4), does not affect the optimal solution.

We show in lemma 3 (appendix B), that the connectivity for the QUBO matrix of the transaction settlement is bounded by twice the average number of transactions per party, $\frac{4I}{K}$, plus a variance term which vanishes for d-regular graphs.

2.2 Qubit-efficient mapping

Mapping QUBO to VQA   The underlying idea to solve QUBO problems with VQAs is to use the PQC to generate bit-vectors $\underline{\mathbf{x}}$. The parameters $\underline{\theta}$ of the PQC are then tuned such that the generated $\underline{\mathbf{x}}$ are likely to approximately minimize equation (1). Formally:

$$\min_{\underline{\theta}} \mathbb{E}_{\underline{\theta}}[C] := \min_{\underline{\theta}} \sum_{\underline{\mathbf{x}} \in \{0,1\}^I} \mathrm{Prob}_{\underline{\theta}}(\underline{\mathbf{x}})\underline{\mathbf{x}}^T Q \underline{\mathbf{x}} = \min_{\underline{\theta}} \sum_{i \neq j} p_{ij}(\underline{\theta})Q_{ij} + \sum_i p_i(\underline{\theta})Q_{ii} \qquad (15)$$

where the right hand side only depends on the marginals $p_{ij}(\underline{\theta}) := \mathrm{Prob}_{\underline{\theta}}(x_i = 1, x_j = 1)$ and $p_i(\underline{\theta}) := \mathrm{Prob}_{\underline{\theta}}(x_i = 1)$.

Instead of searching in a discrete space, this turns the problem into the optimization of the continuous parameters of a generator of bit-vectors. This appears similar but is different from relaxation-based approaches, which often replace binary variables through continuous ones to obtain a more tractable optimization problem whose solutions are projected back to a binary format: Here, the model (the PQC) directly generates bit-vectors $\underline{\mathbf{x}}$ and if it is expressive enough in the distributions $\mathrm{Prob}_{\underline{\theta}}(\underline{\mathbf{x}})$ it parameterizes, then an optimal $\underline{\theta}$ will generate an optimal bit-vector $\underline{\mathbf{x}}$ deterministically. The steps to solve this minimization are shown in Fig. 2.

The core of mapping a QUBO problem to a variational minimization problem therefore consists of specifying how to generate bit-vectors $\underline{\mathbf{x}}$ with a quantum circuit. In standard QAOA or variational approaches, this mapping is straightforward (equation (2)): As the number of qubits n_q equals the number of variables I, we simply measure in the computational basis (Pauli-Z) and associate the outcome 1 (-1) of qubit q_i with the bit x_i equal 0 (1). We use a different mapping, generalizing the qubit-efficient approach in [14]:

Qubit-efficient binary encoding   We use n_a qubits (*ancillas*) to represent a subset of n_a bits and n_r qubits (*register*) to provide an address labelling this subset. Compared to the approach for standard QAOA, we (partly) encode the bit-position in a binary encoded number instead of the one-hot encoded qubit-position. Hence the name "binary encoding" in Table 2.

Formally, consider a covering $\mathcal{A} = \{A_1, \ldots, A_{N_r}\}$, $N_r = 2^{n_r}$ of the set of bit-positions $B = \{1, \ldots, I\}$ with $|A_i| \in \{0, n_a\}$ $\forall i$ and each A_i ordered. Regard the quantum state $|b_1 \ldots b_{n_a}\rangle_{\mathrm{anc}} \otimes |r\rangle_{\mathrm{reg}}$ as corresponding to bit $A_r[l]$ (the lth entry of A_r) equal to b_l $\forall l \in \{1, \ldots, n_a\}$. This quantum state fixes only the subset A_r of the bits. In general, we interpret

- Superpositions in the ancilla state $\leftrightarrow$ probabilistic sampling in the computational basis of different bit-vectors $b_1 \ldots b_{n_a}$
- Superpositions in the register state $\leftrightarrow$ probabilistic sampling in the computational basis of different bit-sets A_r

resulting in the general form

$$|\psi(\underline{\theta})\rangle = \sum_{r=1}^{N_r} \beta_r(\underline{\theta}) \left[a_r^{00\ldots0}(\underline{\theta}) |00\ldots0\rangle_{\mathrm{anc}} + a_r^{00\ldots1}(\underline{\theta}) |00\ldots1\rangle_{\mathrm{anc}} + \cdots \right] \otimes |r\rangle_{\mathrm{reg}}, \qquad (16)$$

where we already indicated that the PQC parameterized by $\underline{\theta}$ determines the values of the register amplitudes ($\beta_r(\underline{\theta})$) and normalized bit-vector amplitudes ($a_r^{b_1 \ldots b_{n_a}}(\underline{\theta})$). We achieve an exponential compression from $n_q = I$ qubits to $n_q = n_a + \lceil \log_2(I/n_a) \rceil$ in the case of a

disjoint covering (also *perfect matching*). In general, a covering consisting of $|\mathcal{A}| = R$ bit-sets requires $n_a + \lceil \log_2(R) \rceil$ qubits.

For the simplest case of the *minimal encoding*, defined by $n_a = 1$, each subset consists of just one binary variable with a total of $n_r = I$ subsets. A quantum state in this encoding can be written as

$$|\psi(\underline{\theta})\rangle_{n_a=1} = \sum_{r=1}^{I} \beta_r(\underline{\theta}) \left[a_r^0(\underline{\theta}) |0\rangle_{\text{anc}} + a_r^1(\underline{\theta}) |1\rangle_{\text{anc}} \right] \otimes |r\rangle_{\text{anc}}, \tag{17}$$

and represents the bit-vector $\underline{\mathbf{x}}$ if $|a_r^i| = \delta_{ix_r}$. The total number of qubits required is $n_q = 1 + \lceil \log_2(I) \rceil$.

The large decrease in qubits comes with a few drawbacks:

- A single measurement in the computational basis only specifies a subset A_r of the bit-positions, and it is not immediate how to sample full bit-vectors $\underline{\mathbf{x}}$.
- Even arbitrary state-preparation through the PQC may only allow limited distributions on the vector x. Consider for example the minimal encoding: It generates bit-vectors distributed as $\text{Prob}_{\underline{\theta}}(\underline{\mathbf{x}}) = \text{Prob}_{\underline{\theta}}^1(x_1) \cdot \ldots \cdot \text{Prob}_{\underline{\theta}}^I(x_I) = \prod_r^I |a_r^1|^2$, where a_r^1 are the coefficients of the ancilla qubits in equation (17), corresponding to a *mean-field* approximation ([81]).
- Different from QAOA, the cost objective may no longer correspond to the expectation of a Hermitian observable. This issue and a resolution are discussed in appendix D.

Sampling algorithm   We will adopt a simple *greedy* approach here, which fixes entries of $\underline{\mathbf{x}}$ as they are sampled throughout multiple measurements and concludes once every entry is sampled. We furthermore determine the covering $\mathcal{A}$ through a k-means-inspired clustering on the graph representation of the problem. Given uniform $\beta_r(\underline{\theta})$ and a disjoint covering $\mathcal{A}$, the probability of any one register not being sampled after n_{shots} measurements is exponentially small, bounded by $\exp(-\frac{n_{\text{shots}}}{N_r})$. In practice, we sample multiple bit-vectors $\underline{\mathbf{x}}$ to find candidates for the optimal solution. This allows us to reduce the average number of measurements by reusing measurement outcomes (in particular those that were sampled multiple times before conclusion of the algorithm). Nonetheless, the qubit compression comes at the cost of significant sampling overhead.

2.3 Cost function

Having specified how to generate bit-vectors from measurement samples of the PQC fully determines $\text{Prob}_{\underline{\theta}}(\underline{\mathbf{x}})$ and hence the minimization problem in equation (15). In practice, we cannot access $\text{Prob}_{\underline{\theta}}(\underline{\mathbf{x}})$ directly, but rather obtain finite-shot measurements on the state prepared by the PQC. Hence, we need to specify an estimator of the expected cost $\mathbb{E}_{\underline{\theta}}[C]$. We will refer to this estimator as $\hat{C}(\underline{\theta})$.

For the explicit formulation of $\hat{C}(\underline{\theta})$, we make use of the formulation of $\mathbb{E}_{\underline{\theta}}[C]$ in terms of marginal probabilities $p_{ij}(\underline{\theta}) = \text{Prob}_{\underline{\theta}}(x_i = 1, x_j = 1)$ and $p_i(\underline{\theta}) = \text{Prob}_{\underline{\theta}}(x_i = 1)$ in equation (15). For the latter, we use heuristic estimators $\hat{p}_i(\underline{\theta})$ and $\hat{p}_{ij}(\underline{\theta})$ which are constructed by counting the number of times a certain bit (or pair of bits) was sampled with value equal to one.

The exact formulas for these estimators are given in appendix C.1 with a derivation for disjoint coverings in appendix C.2. Intuitively,

$$\hat{p}_{ij}(\underline{\theta}) = (1 - \hat{\mu}_{ij})\hat{q}_{ij} + \hat{\mu}_{ij}\hat{p}_i\hat{p}_j \tag{18}$$

where $0 \le \hat{\mu}_{ij} \le 1$ and the asymptotic convergence

$$\hat{p}_i(\underline{\theta}) \xrightarrow{n_{\text{shots}} \to \infty} \frac{\sum_{\substack{r=1 \\ i \in A_r}}^{N_r} |\beta_r(\underline{\theta})|^2 \sum_{\substack{b_k \in \{0,1\} \\ b_{l_r(i)} = 1}} |a_r^{b_1 \dots b_{n_a}}|^2}{\sum_{\substack{r=1 \\ i \in A_r}}^{N_r} |\beta_r(\underline{\theta})|^2} \tag{19}$$

$$\hat{q}_{ij}(\underline{\theta}) \xrightarrow{n_{\text{shots}} \to \infty} \frac{\sum_{\substack{r=1 \\ i,j \in A_r}}^{N_r} |\beta_r(\underline{\theta})|^2 \sum_{\substack{b_k \in \{0,1\} \\ b_{l_r(i)} = b_{l_r(j)} = 1}} |a_r^{b_1 \dots b_{n_a}}|^2}{\sum_{\substack{r=1 \\ i,j \in A_r}}^{N_r} |\beta_r(\underline{\theta})|^2} \tag{20}$$

motivates the expressions $\hat{p}_i$ and $\hat{p}_{ij}$.

Following equation (15), the cost estimator $\hat{C}(\underline{\theta})$ for the transaction settlement problem then takes the form

$$\hat{C}(\underline{\theta}) = \sum_{\substack{i,j=1 \\ i \ne j}}^{I} \hat{p}_{ij}(\underline{\theta})A_{ij} + \sum_{i=1}^{I} \hat{p}_i(\underline{\theta})(A_{ii} + b_i(\underline{\mathbf{s}})) + c(\underline{\mathbf{s}}) \tag{21}$$

which is optimized with respect to $\underline{\theta}$ and the slack variables $\underline{\mathbf{s}} = (\underline{\mathbf{s}}_1, \dots, \underline{\mathbf{s}}_K) \ge 0$. The optimal slack variables can be obtained straightforwardly using

$$\hat{\underline{\mathbf{s}}}_k(\underline{\theta}) = \max\left(\underline{\mathbf{0}}, -\underline{\mathbf{lim}}_k + \sum_{i=1}^{I} \hat{p}_i(\underline{\theta})\underline{\mathbf{v}}_{ik} + \underline{\mathbf{bal}}_k\right) \tag{22}$$

where $\max(\circ, \circ)$ is to be taken element-wise. The optimal slack variables substitute $\underline{\mathbf{s}}$ in equ (21), thus removing the need for separate optimization over the slack variables.

Remarks:

1. It is not possible to express equation (21) as the expectation of a Hermitian observable on a state of the form of equation (16) due to denominators in the expressions for $\hat{p}$, $\hat{q}$ (equ. (54), (56) in appendix) and $\hat{\mu}_{ij}$ (equ. (57)) as well as the functional form of $\hat{\underline{\mathbf{s}}}$. We will show in appendix D how this problem can be resolved for fixed $\underline{\mathbf{s}}$ given uniform $\beta_r(\underline{\theta})$.

2. In the limit of the *full encoding* ($n_a = I$, $n_r = 0$), we get

   $$\hat{p}_{ij} = \hat{q}_{ij} = \left\langle \mathbb{1}_{\min\{i,j\}-1} \otimes |1\rangle\langle 1| \otimes \mathbb{1}_{|i-j|-1} \otimes |1\rangle\langle 1| \otimes \mathbb{1}_{I-i-j}\right\rangle_{\mathcal{M}}$$

   and

   $$\hat{p}_i = \left\langle \mathbb{1}_{i-1} \otimes |1\rangle\langle 1| \otimes \mathbb{1}_{I-i}\right\rangle_{\mathcal{M}}$$

   resulting in the "standard" cost estimator identical to e.g. QAOA.

3. Runtime and memory cost: The naive run-time for classically computing the cost estimator scales as $O(n_{\text{shots}}I^2)$, while the memory required only scales as $O(n_{\text{shots}} + I^2)$. Our approach does not require full tomography with memory requirements as high as $O(4^I)$. In further extensions, methods such as classical shadows ([82]) may be used to more efficiently estimate the cost and reduce n_{shots}.

If certain registers are hardly sampled, i.e. $|\beta_r(\underline{\theta})|^2 \simeq 0$, we may encounter division by zero in the expressions for $\hat{p}_i$ and $\hat{q}_{ij}$. In practice, this can be dealt with by setting the corresponding estimators to $1/2$ whenever estimates for $|\beta_r(\underline{\theta})|^2$ fall below some $\epsilon > 0$, resulting in indirect penalization. Alternatively, we can add an explicit regularization term $\hat{R}(\underline{\theta}) = \eta \sum_{r=1}^{N_r} \left[\hat{r}_r(\underline{\theta}) - \frac{1}{N_r} \right]^2$ to the cost function.

2.4 Variational ansatz

In this work, we consider two types of PQC:

- A *hardware-efficient* ansatz consisting of RY rotations and entangling CNOT layers.
- A *register-preserving* ansatz of conditional RY rotations incorporating constraints and symmetries tailored to the qubit-efficient encoding.

Both variational circuits are depicted in Fig. 3. The hardware-efficient ansatz was used identically in [14], the register-preserving ansatz is one of the main contributions of this work.

We mentioned difficulties arising from vanishing register-amplitudes in the previous section. We will now formally define register-uniform quantum states and register-preserving circuits before discussing the advantages offered by them:

Definition 1 (Register-uniform) We call a quantum state $|\psi\rangle_{ar} \in \mathcal{H}_{\text{anc}} \otimes \mathcal{H}_{\text{reg}}$ *register-uniform* with respect to the orthonormal basis $(|1\rangle, \ldots, |N_r\rangle)$ of $\mathcal{H}_{\text{reg}}$, if it can be written as
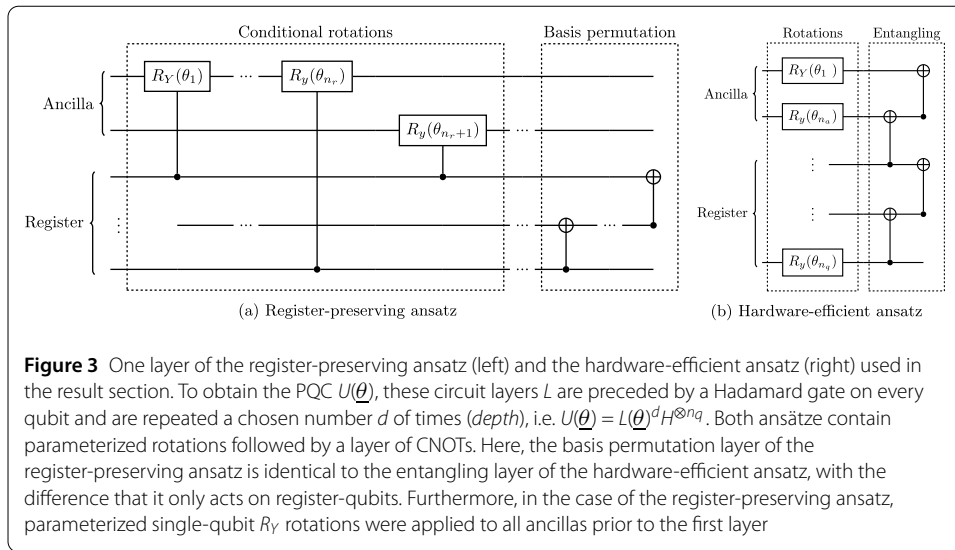
$$|\psi\rangle = \frac{1}{\sqrt{N_r}} \sum_{r=1}^{N_r} |\phi_r\rangle_{\text{anc}} \otimes |r\rangle_{\text{reg}} \tag{23}$$

where $|\phi_r\rangle_{\text{anc}} \in \mathcal{H}_{\text{anc}}$ is arbitrary with $\langle \phi_r | \phi_r \rangle_{\text{anc}} = 1 \; \forall r$.

Definition 2 (Register-preserving) We call a unitary U acting on $\mathcal{H}_{\text{anc}} \otimes \mathcal{H}_{\text{reg}}$ *register-preserving* with respect to the orthonormal basis $(|1\rangle, \ldots, |N_r\rangle)$ of $\mathcal{H}_{\text{reg}}$ if it always maps register-uniform states to register-uniform states (with respect to the same basis).

The set of register-preserving unitaries with respect to the same basis is closed under concatenation. Our register-preserving ansatz first prepares the register-uniform plus state $|+\rangle^{\otimes n_q} = H^{\otimes n_q} |0\rangle$ (H being the Hadamard gate) and then acts through register-preserving unitaries on it.

Notation   We use the bra-/ket-notation only for normalized states. Furthermore, Latin letters inside bra and ket indicate computational basis states, while Greek letters indicate general quantum states. We will refer to *register-preserving circuits* as quantum circuits which output register-uniform states with respect to the computational basis (and fix said basis from now on, omitting further mention of it).

Figure 3 One layer of the register-preserving ansatz (left) and the hardware-efficient ansatz (right) used in the result section. To obtain the PQC $U(\underline{\theta})$, these circuit layers L are preceded by a Hadamard gate on every qubit and are repeated a chosen number d of times (*depth*), i.e. $U(\underline{\theta}) = L(\underline{\theta})^d H^{\otimes n_q}$. Both ansätze contain parameterized rotations followed by a layer of CNOTs. Here, the basis permutation layer of the register-preserving ansatz is identical to the entangling layer of the hardware-efficient ansatz, with the difference that it only acts on register-qubits. Furthermore, in the case of the register-preserving ansatz, parameterized single-qubit R_Y rotations were applied to all ancillas prior to the first layer

The following claims about register-uniform states and register-preserving circuits are proved in appendix B:

Lemma 1 *The following is equivalent to a state $|\psi\rangle \in \mathcal{H}_{anc} \otimes \mathcal{H}_{reg}$ being register-uniform*:

$$\langle r| \, tr_{\mathcal{H}_{anc}}(|\psi\rangle \langle\psi|) |r\rangle = \frac{1}{N_r} \; \forall r \in \{1,\dots,N_r\} \tag{24}$$

Theorem 1 *The following are equivalent for a unitary U acting on $\mathcal{H}_{anc} \otimes \mathcal{H}_{reg}$:*
  (i) *U is register-preserving*
 (ii) *$U(|\phi\rangle_{anc} \otimes |r\rangle_{reg}) = (U_r |\phi\rangle_{anc}) \otimes |f(r)\rangle$ where U_r is a unitary on $\mathcal{H}_{anc}$ $\forall r$ and $f : \{1,\dots,N_r\} \to \{1,\dots,N_r\}$ is bijective.*
(iii) *U can be written as a sequence of unitary matrices on $\mathcal{H}_{anc}$ conditioned on a subset of register-qubits and basis-permutations on the register.*

Note,[9] that in theorem 1 we refer to *basis*-permutations, not *qubit*-permutations, although former contains the latter.

Theorem 1 provides a list of ingredients that may be used to construct register-preserving variational ansätze. Namely, we can combine conditional unitaries (such as CNOT, Toffoli gates), acting on the ancillas and conditioned on the register-qubits, with arbitrary unitaries that only act on the ancilla qubits. Furthermore, we can permute computational basis states on the register qubits. These permutations could be cryptographic permutation pads [83], binary adder circuits [84] or heuristic constructions from NISQ-friendly gates such as CNOT, SWAP and X gates.

When defining register-preserving circuits, we demand that *any* register-uniform state is mapped to a register-uniform state. This may not always be necessary. In the case of this work, we always start with the same input state $|+\rangle^{\otimes n_q}$ which allows more general unitaries than theorem 1, as the following lemma demonstrates:

---

[9]To avoid confusion with the *permutation operator* used in quantum physics which refers to permuting particle-labels.

Lemma 2 *For register-uniform states as in equation* (23) *with* $\langle\phi_{r_1}|\phi_{r_2}\rangle_{anc} \in \mathbb{R} \ \forall r_1, r_2 \in \{1,\ldots,N_r\}$, *a unitary* $U = \mathbb{1}_{anc} \otimes U_{reg}$ *only non-trivially acting on the register-qubits always maps* $|\psi\rangle$ *to a register-uniform state if and only if*

$$(U^\dagger)_{r_1 s} U_{s r_2} \in i\mathbb{R} \ \forall r_1, r_2, s \in \{1,\ldots,N_r\} \ with \ r_1 < r_2. \tag{25}$$

While theorem 1 only allows permutations on the register-qubits, this lemma allows (a single) application of $\exp\left(-i\frac{\theta}{2}P\right)$ for any self-inverse permutation P, e.g. RX^n rotations or the RBS gate, on the register-qubits. The condition $\langle\phi_{r_1}|\phi_{r_2}\rangle_{anc} \in \mathbb{R}$ is trivially fulfilled for states which are a real-valued linear combination of computational basis states such as $|+\rangle^{\otimes n_q}$.
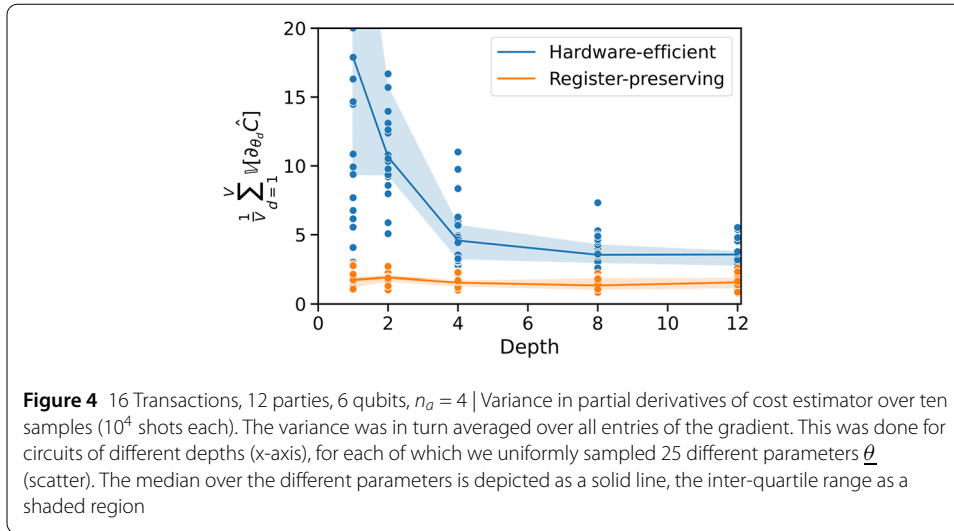
Our ansatz   The circuits used in this work are shown in Fig. 3. The register-preserving circuit acts with conditional RY rotations on every ancilla-qubit, conditioned on individual register-qubits. The RY rotation on ancilla-qubit b conditioned on register-qubit c can be regarded as a parameterized rotation on b for half the registers (those registers r, for which the binary encoding of r has a 1 at position c). A basis permutation layer consisting of CNOTs is added to ensure consecutive conditional RY rotations act on a different set of registers (this basis permutation layer is omitted if only a single layer is used, $d = 1$).

In terms of optimization parameters, we optimize $n_a * n_r$ parameters per register-preserving layer and $n_q = n_r + n_a$ for the hardware-efficient ansatz.

Discussion register-preserving ansatz   Only allowing register-preserving gates in the variational ansatz imposes challenges in keeping the variational ansatz both expressive and NISQ-friendly, at least on superconducting hardware (cf. Sect. 3.2). On the other hand, we see the following motivations and advantages for exploring register-preserving circuits:

1. Respect the symmetries of the qubit-efficient approach: In light of challenges associated with barren plateaus for over-expressive ansätze, incorporating symmetries into the circuit architecture – here: register-preservation and real-valued amplitudes[10] in the computational basis – is promising as it has been shown to help with the problem of vanishing gradients ([40, 80]).

2. Numeric stability: The cost estimator $\hat{C}(\underline{\theta})$ (equ. (21)) makes use of estimators for the register-amplitudes $|\beta(\underline{\theta})|^2$. These can be fixed to $\frac{1}{N_r}$ for a register-preserving circuit, adding numerical stability (especially as the terms affected are in the denominator) and reducing the computational overhead. In Fig. 4, we visualize the variance of the gradient-estimator with respect to shot noise at fixed parameters $\underline{\theta}$. The register-preserving ansatz shows much smaller variance, which suggests a lower number of required shots n_{shots} (cf. bullet 3.).

3. Sampling overhead: Register-uniform states minimize the expected number of samples needed to cover each register ([85]). Furthermore, theorem 1 (*ii*) shows, that the net effect of any register-preserving unitary on the register-qubits is a permutation. If this permutation $P : |r\rangle \mapsto |f(r)\rangle$ is easily inverted, then the bit-vector sampling can be made deterministic in the register (without otherwise impacting the prediction), by using the input state $H^{\otimes n_a} |0\rangle_{anc} \otimes P^{-1} |r\rangle_{reg}$ instead of

---

[10]Real-valued amplitudes are the reason we only make use of RY rotations (instead of RX, RZ).

Figure 4 16 Transactions, 12 parties, 6 qubits, $n_a = 4$ | Variance in partial derivatives of cost estimator over ten samples (10^4 shots each). The variance was in turn averaged over all entries of the gradient. This was done for circuits of different depths (x-axis), for each of which we uniformly sampled 25 different parameters $\underline{\theta}$ (scatter). The median over the different parameters is depicted as a solid line, the inter-quartile range as a shaded region

$H^{\otimes n_a + n_r} |0\rangle$. In any case, we can reduce the number of circuit evaluations needed by using initial states of the form $H^{\otimes n_a} |0\rangle_{\text{anc}} \otimes |r\rangle_{\text{reg}}$ and ensuring that we sample a different register $f(r)$ in every run by iterating over $r \in \{1, \ldots, N_r\}$.

4. Expression as expectation value of Hermitian observable: As all denominators in the expressions for $\hat{p}_i$, $\hat{q}_{ij}$ and all of $\hat{\mu}$ are replaced by constants, this allows – for fixed $\underline{s}$ – to express $\hat{C}(\underline{\theta})$ as the expectation value of a Hermitian observable (although the product $\hat{p}_i \hat{p}_j$ requires preparation of a product state, see appendix D). A majority of the literature (including aforementioned classical shadows) and software are tailored primarily for Hermitian expectation values. Areas include theoretical results (e.g. adiabatic theorem), the variational ansatz and optimizer itself, estimation and error mitigation as well as fault-tolerant methods for the evaluation of expectation values. Expressing our cost function as a Hermitian expectation hence widens the cross-applicability of other results and code-bases.

2.5 Optimization

Many different optimization procedures have been suggested in the literature to find the optimal parameters for a PQC through classical optimization. This includes the parameter-initialization ([36, 86–89]), choice of meta-parameters ([90]) as well as the parameter update itself ([38, 91–94], an overview of gradient-based and gradient-free optimizers can be found in section D. of [12]).

While the optimization of a PQC has been shown to be NP-hard ([95]) and may well be the most important ingredient to practical advantage for any quantum QUBO solver, the focus of this work is on the qubit-efficient methods rather than on the optimization itself. Our results were obtained with two different commonly used optimizers: The gradient-free optimizer COBYLA (implemented in scipy [96]) as well as standard gradient descent, with gradients calculated through the parameter-shift rule ([92, 97]).

The full optimization step for updating the circuit parameters $\underline{\theta}^{(n)} \mapsto \underline{\theta}^{(n+1)}$ is depicted in Fig. 2.

Figure 5 16 Transactions, 10 parties, 5 (register-efficient) / 16 (QAOA) qubits, $n_a = 1$, 2×10^4 shots | Cost during parameter optimization for different circuit ansätze (Hardware-efficient, Register-preserving and QAOA) applied to a transaction settlement problem with 16 transactions. Two different optimizers, COBYLA (gradient-free) and gradient descent (DESC) were used in the qubit-efficient approach while only COBYLA was used for QAOA. The depth of the circuit (in number of layers/p-value for QAOA) is given in square brackets. The solid lines depict means over 25 (qubit-efficient) and 10 (QAOA) different training runs with random starting points. The 95% confidence interval is given as a shaded region. Quantitative values against QAOA are not comparable due to different cost functions used for the parameter training

3 Results and discussion

Here, we present results from applying the methodology presented in Sect. 2 to transaction settlement problems of 16 and 128 transactions. We compare hardware-efficient and register-preserving qubit compression with QAOA. We show results for both a simulator backend (Pennylane [98]) and quantum hardware from IBM Quantum and IonQ. The statistics for uniformly random solution-sampling is also provided for benchmarking. For 16 transactions, this includes the optimal solution. Throughout this section R (see 2.1) was set to $\lfloor \frac{I}{4} \rfloor$ and we considered only cash and one security ($J = 2$). We randomly generated three sets of $I = 16$ transaction instructions with $K = 10$, 12 and 13 parties respectively and one settlement problem with 128 transaction instructions, $K = 41$.

3.1 Simulation, 16 transactions – comparison with QAOA

Training convergence   Figure 5 shows the training convergence during the parameter optimization, averaged over different random initial parameters of the PQC. While COBYLA returned optimized parameters within a few hundred steps or less, its cost value is consistently outperformed by gradient descent (DESC), especially for an increasing number of circuit parameters.

The register-preserving ansatz not only outperforms the hardware-efficient PQC, but also produces solutions with less variance for different starting points. For all the qubit-efficient approaches, deeper circuits also improved the performance.

For QAOA, the substitution in equ (22) to optimize both slack variables and variational parameters simultaneously is infeasible as the variational ansatz depends on the QUBO matrix and by extension, the slack variables (cf. equation (6)). Results for QAOA were obtained by alternating the optimization of slack variables and circuit parameters 50 times,

with up to 10^3 COBYLA-iterations to update the circuit parameters at each cycle. The optimization landscape appears to be dominated by the slack variables, and each update changes the optimization landscape for the variational parameters. This unusual optimization landscape is likely the reason why no significant improvements were observed for increasing p-values.
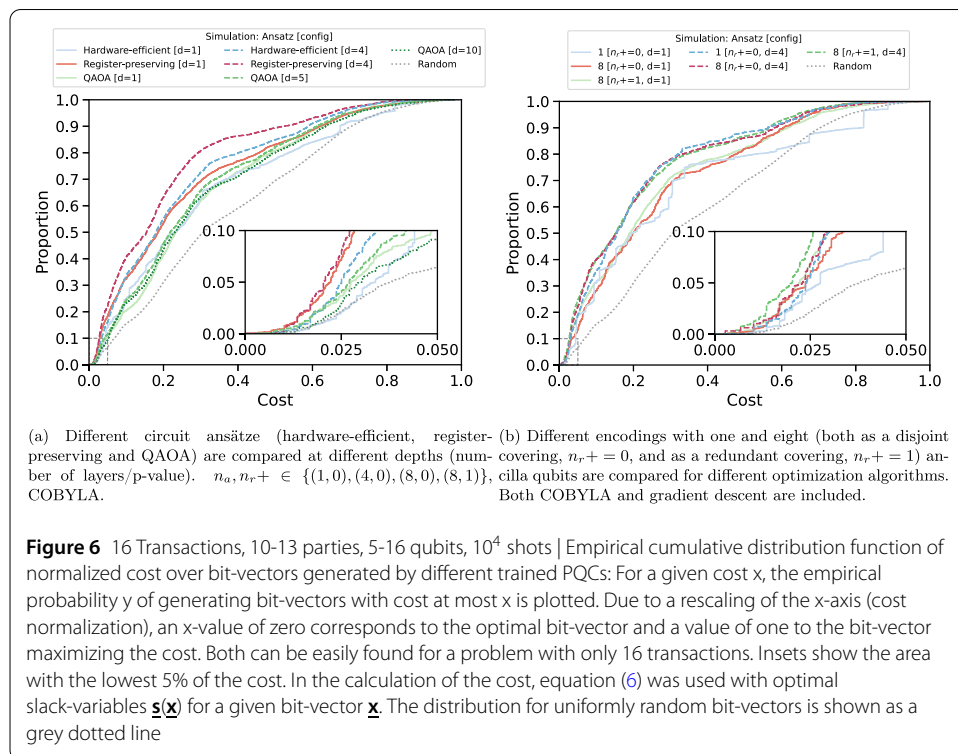
From our brief comparison, hardware-efficient ansätze appear to be more suited for MBO problems as they are agnostic to changes in the QUBO matrix. Despite these challenges, we maintain our QAOA results for the purposes of comparison and leave the exploration of more effective implementations of the QAOA to MBO to future work.

Bit-vector quality   As the cost estimator used in the optimization is only a proxy for the actual quality of the bit-vectors $\underline{x}$ generated, we show the empirical cumulative distribution of the cost associated with bit-vectors generated from the trained PQC in Fig. 6. We normalized the cost for each transaction settlement problem and averaged over different configurations (three settlement problems, $n_a \in \{1, 4, 8\}$ for 6a, COBYLA and gradient descent, up to 25 training runs), drawing 50 bit-vectors per configuration.

Subfigure 6a shows the cumulative distributions for both the qubit-efficient approach and QAOA. Except for the hardware-efficient ansatz with one layer, our qubit-efficient approach performs better than QAOA on average. As in the training traces, no significant differences in the results for QAOA were found by varying the depth (p-value) from one to ten. The register-preserving ansatz performs best for all depths.

In subfigure 6b, weak improvement can be observed by using 8 instead of 1 ancilla qubits and by adding another register-qubit ($n_r+ = 1$).

During the training, we observed that gradient descent yields better minima than COBYLA in the cost estimator but tends to *sparse* solutions, i.e. the associated distri-

(a) Different circuit ansätze (hardware-efficient, register-preserving and QAOA) are compared at different depths (number of layers/p-value). $n_a, n_r+ \in \{(1,0),(4,0),(8,0),(8,1)\}$. COBYLA.

(b) Different encodings with one and eight (both as a disjoint covering, $n_r+ = 0$, and as a redundant covering, $n_r+ = 1$) ancilla qubits are compared for different optimization algorithms. Both COBYLA and gradient descent are included.

Figure 6 16 Transactions, 10-13 parties, 5-16 qubits, 10^4 shots | Empirical cumulative distribution function of normalized cost over bit-vectors generated by different trained PQCs: For a given cost x, the empirical probability y of generating bit-vectors with cost at most x is plotted. Due to a rescaling of the x-axis (cost normalization), an x-value of zero corresponds to the optimal bit-vector and a value of one to the bit-vector maximizing the cost. Both can be easily found for a problem with only 16 transactions. Insets show the area with the lowest 5% of the cost. In the calculation of the cost, equation (6) was used with optimal slack-variables $\underline{s}(\underline{x})$ for a given bit-vector $\underline{x}$. The distribution for uniformly random bit-vectors is shown as a grey dotted line

bution on bit-vectors is strongly concentrated around a single value (cf. Fig. 7). Here, redundant encoding of bit-vector-positions in the ancillas (i.e. $n_r+ > 0$) was found to help in generating more diverse solution candidates.

3.2 Hardware, 16 & 128 transactions – results on IonQ and IBMQ QPUs

To investigate the generation of bit-vectors on real quantum hardware (QPU), we optimized different configurations of both register-preserving and hardware-efficient PQCs on a simulator for 16 and 128 transactions. The pre-trained circuit parameters were then executed on the Geneva/Hanoi QPU provided by IBM Quantum and the Harmony/Aria QPU provided by IonQ.[11] The resulting cost-distributions of generated bit-vectors for a settlement problem with 16 transactions and 10 parties / 128 transactions and 41 parties are depicted in Fig. 7 and 8.

IBMQ vs IonQ　　For 4 layers of the register-preserving circuit (7a), the IBMQ results are significantly worse than for IonQ. This can partly be attributed to the connectivity requirements of the long-range conditional Y-Rotations used in the register-preserving ansatz (Fig. 3a). This favours the all-to-all connectivity of `ionq_harmony`, which forgoes the need for depth-increasing SWAP networks. The hardware-efficient ansatz (7b) on the other hand is compatible with the lattice connectivity of IBMQ devices and shows similar performance for both QPUs.

For 128 transactions, the benchmarked IonQ device (`ionq_aria`) slightly outperforms IBMQ (`ibm_hanoi`) even with the hardware-efficient ansatz, the results of which are depicted in Fig. 8.

Impact of noise　　In general, the noisy results obtained from real quantum backends yielded worse results than noise-free simulations. However, the additional variation in the generated bit-vectors could also help to generate solutions of lower cost. This is observed in Fig. 8a, where the hardware results yielded bit-vectors of lower cost than the lowest simulated vectors with a probability of more than 10%. Noise does not necessarily move the distribution towards uniform random bit-vectors: On real hardware, the decay into the physical ground state is more likely than the excited state. Depending on the σ_z-to-bit mapping, this can result in a larger or smaller number of settled transactions than uniform randomness and, potentially, in performance worse than uniform random (as for `ibm_geneva` in Fig. 7a).

Figure 8b emphasizes the need for classical post-processing methods that search for feasible solutions in the vicinity of infeasible solutions generated by the PQC (cf. 4): None of the bit-vectors generated by both simulation and real hardware fulfil all the constraints on the security-account balances (cf. equation (4)). Alternatively, the cost penalty λ could be increased to put even higher priority on the balance constraints relative to the maximization of the number of transactions.

4 Conclusion and outlook

Increasing the scope of tractable problems and benchmarking with industry data is important to gauge the applicability of heuristics-reliant variational quantum algorithms to optimization and to find promising applications. In this work, we extended the qubit-efficient

---

[11]Different backends from both providers were used as `ibmq_geneva` was retired while this work was in process and `ionq_harmony` only provides 11 qubits, necessitating the larger `ionq_aria` device for 128 transactions with 16 ancillas.

(a) Register-preserving ansatz   (b) Hardware-efficient ansatz

Figure 7 16 Transactions, 10 parties, 6 qubits, $d = 4$, DESC, $n_a = 4$, disjoint covering | Empirical cumulative distribution function (cf. Fig. 6) of cost (not normalized) over bit-vectors generated by a trained PQC. Executed on a simulator (Simulation), IBMQ quantum computer (`ibm_geneva`) and IonQ quantum computer (`ionq_harmony`). Uniformly random bit-vectors are shown by a grey line (Random). For each generator, 1000 bit-vectors were drawn from 2.4×10^4 shots. Each subfigure corresponds to a single optimization run in which the PQC was optimized using gradient descent. Both runs produce narrow distributions around a fixed bit-vector, resulting in a steep curve (especially in the noise-free simulation)

(a) Empirical cumulative distribution function (cf. figure 6). To generate the blue dotted line, showing the distribution of uniformly random bit-vectors, we generated 10^6 such vectors, the cost of which was used to normalize the x-axis.

(b) Scatter-plot of sampled bit-vectors. Scatter-size increases with degeneracy. The x-axis shows the negative sum of final security-account balances over all parties with negative balances upon settlement according to the corresponding bit-vector. The number of transactions is the L_1 norm of the corresponding bit-vector.

Figure 8 128 transactions, 41 parties, 19 qubits, hardware-efficient ansatz, $d = 1$, DESC, $n_a = 16$, disjoint covering | Visualization of results for 128 transactions on the `ibm_hanoi` and `ionq_aria` backends (2.4×10^4 shots, 500 bit-vectors)

encoding in [14] by providing explicit formulas of the cost objective and its gradient for arbitrary number of ancilla qubits. We introduced a new ansatz for uniform register sampling. We argue that register-preserving ansätze have the additional benefits of numerical stability, shot-reductions and selective sampling of individual registers, and expressing the cost estimator as a Hermitian observable.

We demonstrate our methods on mixed binary optimization problems arising from financial transaction settlement [19], benchmarking problems of up to 128 transactions

and 41 parties constructed from transaction data provided by a regulated financial exchange. We also showed how the optimal slack variables can be obtained without the need for an outer loop optimization. We observed that our qubit-efficient methods outperformed standard QAOA, even when executed on quantum backends. Our proposed register-preserving ansatz stood out as best in many of the instances considered.

Post-processing   While not explored in this work, post-processing by projecting sampled solutions to valid bit-vectors fulfilling all problem constraints may be a necessity for practical use. One possible method to do so includes projecting a solution bit-vector generated by the QS to the best bit-vector *in the vicinity* that fulfils all constraints. Restricting the search to a ball of Hamming distance $\leq k$, solutions can be sampled at the asymptotic runtime of $O(I^k)$ (as opposed to 2^I for a full search). While without guarantees for the optimality or even existence of a close valid solution, one may hope that if the QS provides solutions of high quality, only small adjustments are needed to obtain a good solution which adheres to all constraints. This search can be refined with heuristics, e.g. by only adjusting transactions involving parties (and potentially their k-nearest-neighbours on the graph) whose balance constraints are violated.

Method exploration   Overall, further exploration of different ancilla-register-mappings, variational (register-preserving) ansätze, optimization algorithms and (scaling of) meta-parameters such as circuit depth, penalty terms and step-size is warranted to validate and refine the qubit-compression approach presented in this work. For example, how the restrictions given by theorem 1 regarding the register-preserving ansatz can best be extended in practice, e.g. by changing the computational basis and keeping track of phases on the ancilla qubits (cf. 2), is still an open question. Another important consideration is the lack of correlation between the individual measurements in the sampling algorithm used in this work. Exploring sample rejection or the addition of hidden layers to the ansatz provides one avenue to extend the probability distributions of $\underline{x}$ which our PQC parameterizes.

Our methods, despite being tested on synthetic problems, demonstrated the broad applicability of quantum algorithms beyond small toy examples. Witnessing advantages of our methods over classical solvers would require a comparison to state-of-the-art classical solvers on problem instances faced in real scenarios. Most of our methods are directly applicable beyond the transaction settlement problem to any QUBO problem with linear inequality constraints, setting them apart from other qubit-efficient methods to the best of our knowledge and making them suitable to tasks beyond settling financial transactions.[12]

---

[12]For a list of examples see [20] ch. 2.2(.1).

Appendix A: Additional figures

Figure 9 128 Transactions (each with security and optional cash payment), 41 parties | Parties and transactions are shown similar to Fig. 1 for a settlement problem generated from data of a regulated financial exchange (same settlement problem as Fig. 8). Transfers of securities (cash) are visualized by dotted (solid) edges and green (blue) arrow-heads. The thickness of the edges scales proportional to the transaction volume. The nodes represent parties and are colored and scaled according to the number of in- and outgoing transfers. The edge-coloring corresponds to the register-mapping, i.e. the sets $A_i \in \mathcal{A}$. For $n_a = 16$ ancilla qubits and $n_r = 3$ register qubits used to generate the register-mapping, we have $2^3 = 8 = |\mathcal{A}|$ different colors

Appendix B: Proofs

In this section, we provide formal proofs of claims made in the main text:

Register-preserving ansatz.

Lemma 1 *The following is equivalent to a state $|\psi\rangle \in \mathcal{H}_{anc} \otimes \mathcal{H}_{reg}$ being register-uniform*:

$$\langle r| tr_{\mathcal{H}_{anc}}(|\psi\rangle \langle \psi|)|r\rangle = \frac{1}{N_r} \, \forall r \in \{1, \ldots, N_r\} \tag{26}$$

Proof " $\Longrightarrow$ ": Given a register-uniform state $|\psi\rangle = \frac{1}{\sqrt{N_r}} \sum_{r=1}^{N_r} |\phi_r\rangle_{\text{anc}} \otimes |r\rangle_{\text{reg}}$, direct calculation shows

$$\langle r| tr_{\mathcal{H}_{\text{anc}}}(|\psi\rangle \langle \psi|)|r\rangle_{\text{reg}} = \frac{1}{N_r} tr(|\phi_r\rangle \langle \phi_r|_{\text{anc}}) = \frac{1}{N_r}. \tag{27}$$

" $\Longleftarrow$ ": Given $|\psi\rangle$ with $\langle r| tr_{\mathcal{H}_{\text{anc}}}(|\psi\rangle \langle \psi|)|r\rangle = \frac{1}{N_r}$, write $|\psi\rangle$ in the computational basis as

$$|\psi\rangle = \sum_{i=1}^{N_a} \sum_{r=1}^{N_r} \lambda_{ir} |i\rangle_{\text{anc}} \otimes |r\rangle_{\text{reg}} = \frac{1}{\sqrt{N_r}} \sum_{r=1}^{N_r} \underbrace{\left[\sqrt{N_r} \sum_{i=1}^{N_a} \lambda_{ir} |i\rangle_{\text{anc}} \right]}_{=:|\phi_r\rangle_{\text{anc}}} \otimes |r\rangle_{\text{reg}} \tag{28}$$

Then

$$\langle \phi_r|\phi_r\rangle_{\text{anc}} = tr(|\phi_r\rangle \langle \phi_r|_{\text{anc}}) = N_r tr(\mathbb{1} \otimes \langle r|_{\text{reg}} |\psi\rangle \langle \psi| \mathbb{1} \otimes |r\rangle_{\text{reg}})$$

$$= N_r \langle r| \operatorname{tr}_{\mathcal{H}_{\mathrm{anc}}}(|\psi\rangle \langle\psi|) |r\rangle = 1 \tag{29}$$

showing that $|\psi\rangle$ is register-uniform. $\square$

Theorem 1 *The following are equivalent for a unitary U acting on $\mathcal{H}_{anc} \otimes \mathcal{H}_{reg}$:*

(i) *U is register-preserving*

(ii) *$U(|\phi\rangle_{anc} \otimes |r\rangle_{reg}) = (U_r |\phi\rangle_{anc}) \otimes |f(r)\rangle$ where U_r is a unitary on $\mathcal{H}_{anc}$ $\forall r$ and $f : \{1,\ldots,N_r\} \rightarrow \{1,\ldots,N_r\}$ is bijective.*

(iii) *U can be written as a sequence of unitary matrices on $\mathcal{H}_{anc}$ conditioned on a subset of register-qubits and basis-permutations on the register.*

Proof We will proof in order $(i) \implies (ii) \implies (iii) \implies (i)$.

$(i) \implies (ii)$: The idea in this part is the following: Since the ancilla states for each register are arbitrary, no "mixing" between different registers is allowed as this would result in uncontrollable superpositions on the ancilla subsystems causing a loss of "normalization". We will first show the following weaker claim:

Claim 1: $\forall |\phi_r\rangle_{\mathrm{anc}} \in \mathcal{H}_{\mathrm{anc}}, 1 \le r \le N_r, \exists |\tilde{v}_r\rangle_{\mathrm{anc}} \in \mathcal{H}_{\mathrm{anc}}$ and $1 \le \tilde{r} \le N_r$, s.t. $U |\phi_r\rangle_{\mathrm{anc}} |r\rangle_{\mathrm{reg}} = |\tilde{v}_r\rangle_{\mathrm{anc}} |\tilde{r}\rangle_{\mathrm{reg}}$

Proof of Claim 1: Assume this was not the case, i.e. $\exists |\phi_s\rangle_{\mathrm{anc}} \in \mathcal{H}_{\mathrm{anc}}, 1 \le s \le N_r$ s.t.

$$U |\phi_s\rangle_{\mathrm{anc}} |s\rangle_{\mathrm{reg}} = \lambda |\tilde{v}_s\rangle_{\mathrm{anc}} |\tilde{s}\rangle_{\mathrm{reg}} + \bar{\lambda} |\tilde{\psi}\rangle \tag{30}$$

for some $\lambda, |\tilde{v}_s\rangle_{\mathrm{anc}}, \tilde{s}, |\tilde{\psi}\rangle$ with $0 < |\lambda| < 1$ and $\mathbb{1}_{\mathrm{anc}} \otimes \langle\tilde{s}|_{\mathrm{reg}} |\tilde{\psi}\rangle = 0$. We denoted $\bar{\lambda} := \sqrt{1 - |\lambda|^2}$. We note at this point, that adding a complex phase to $|\phi_s\rangle_{\mathrm{anc}}$ merely adds the same phase to λ and $|\tilde{\psi}\rangle$ but otherwise does not change equation (30).

We now apply U to a register-uniform state containing $|\phi_s\rangle_{\mathrm{anc}} |s\rangle_{\mathrm{reg}}$, resulting in

$$|\tilde{\phi}_{\tilde{s}}\rangle_{\mathrm{anc}} := \mathbb{1}_{\mathrm{anc}} \otimes \langle\tilde{s}|_{\mathrm{reg}} U \sum_{r=1}^{N_r} |\phi_r\rangle_{\mathrm{anc}} \otimes |r\rangle_{\mathrm{reg}} \tag{31}$$

$$= \lambda |\tilde{v}_s\rangle_{\mathrm{anc}} + \mu |\tilde{\eta}\rangle_{\mathrm{anc}} \tag{32}$$

where $\mu |\tilde{\eta}\rangle_{\mathrm{anc}} = \mathbb{1}_{\mathrm{anc}} \otimes \langle\tilde{s}|_{\mathrm{reg}} U \sum_{\substack{r=1 \\ r \ne s}}^{N_r} |\phi_r\rangle_{\mathrm{anc}} \otimes |r\rangle_{\mathrm{reg}}$. We will fix $|\phi_r\rangle_{\mathrm{anc}}$ for $r \ne s$ later. Using that U is register-preserving, we have

$$1 = \langle\tilde{\phi}_{\tilde{s}}|\tilde{\phi}_{\tilde{s}}\rangle_{\mathrm{anc}} = |\lambda|^2 + |\mu|^2 + \lambda^* \mu \langle\tilde{v}_s|\tilde{\eta}\rangle_{\mathrm{anc}} + \lambda \mu^* \langle\tilde{\eta}|\tilde{v}_s\rangle_{\mathrm{anc}} . \tag{33}$$

If we can show that $\mu \langle\tilde{v}_s|\tilde{\eta}\rangle_{\mathrm{anc}} \in \mathbb{R}_{>0}$, then we can choose the phase of $|\phi_s\rangle_{\mathrm{anc}}$ such that

- Case 1: $\lambda = |\lambda|$
- Case 2: $\lambda = i|\lambda|$

which results in the contradiction

$$|\lambda|^2 + |\mu|^2 + 2|\lambda|\mu \langle\tilde{v}_s|\tilde{\eta}\rangle_{\mathrm{anc}} = 1 = |\lambda|^2 + |\mu|^2. \tag{34}$$

It remains to show, that we can choose $|\phi_r\rangle_{\mathrm{anc}}, r \ne s$ such that $\mu \langle\tilde{v}_s|\tilde{\eta}\rangle_{\mathrm{anc}} \ne 0$ and real.

For this consider

$$U^{-1} |\tilde{v}_s\rangle_{\text{anc}} |\tilde{s}\rangle_{\text{reg}} = \frac{1}{\lambda} |\phi_s\rangle_{\text{anc}} |s\rangle_{\text{reg}} - \frac{\bar{\lambda}}{\lambda} U^{-1} |\tilde{\psi}\rangle \tag{35}$$

Claim 2: $\exists w \neq s \in \{1, \dots, N_r\} : \mathbb{1}_{\text{anc}} \otimes \langle w|_{\text{reg}} U^{-1} |\tilde{v}_s\rangle_{\text{anc}} |\tilde{s}\rangle_{\text{reg}} \neq 0$
Proof of Claim 2: Assume this was not the case. Then

$$U^{-1} |\tilde{v}_s\rangle_{\text{anc}} |\tilde{s}\rangle_{\text{reg}} = \frac{1}{\lambda} |\phi_s\rangle_{\text{anc}} |s\rangle_{\text{reg}} + \sigma |\theta_s\rangle_{\text{anc}} |s\rangle_{\text{reg}} \tag{36}$$

Hence $\text{span}\{|\tilde{v}_s\rangle_{\text{anc}} |\tilde{s}\rangle_{\text{reg}}\} \cap \text{Im}[U_{|\text{span}\{\mathcal{H}_{\text{anc}} \otimes |r\rangle_{\text{reg}}\}}] = \emptyset \ \forall r \neq s$. In particular for $r \neq s$

$$\dim \left(\text{Im}[U_{|\text{span}\{\mathcal{H}_{\text{anc}} \otimes |r\rangle_{\text{reg}}\}}] \cap \text{span}\{\mathcal{H}_{\text{anc}} \otimes |\tilde{s}\rangle_{\text{reg}}\} \right)$$
$$\leq \dim(\mathcal{H}_{\text{anc}}) - 1 = \dim(\text{Im}[U_{|\text{span}\{\mathcal{H}_{\text{anc}} \otimes |r\rangle_{\text{reg}}\}}]) - 1. \tag{37}$$

Therefore

$$\forall r \neq s \ \exists |\phi_r\rangle_{\text{anc}} \ \text{s.t} \ \mathbb{1}_{\text{anc}} \otimes \langle \tilde{s}|_{\text{reg}} U |\phi_r\rangle_{\text{anc}} \otimes |r\rangle_{\text{reg}} = 0 \tag{38}$$

which implies

$$\mathbb{1}_{\text{anc}} \otimes \langle \tilde{s}|_{\text{reg}} U \sum_{r=1}^{N_r} |\phi_r\rangle_{\text{anc}} \otimes |r\rangle_{\text{reg}} = \lambda |\tilde{v}_s\rangle_{\text{anc}} |\tilde{s}\rangle_{\text{reg}} \tag{39}$$

which is not normalized ($|\lambda| < 1$), in contradiction with U being register-preserving.
Therefore claim 2 holds and it exists $|\phi_w\rangle_{\text{anc}} \otimes |w\rangle_{\text{reg}}$ with $w \neq s$ such that

$$\langle \tilde{v}_s|_{\text{anc}} \otimes \langle \tilde{s}|_{\text{reg}} U |\phi_w\rangle_{\text{anc}} \otimes |w\rangle_{\text{reg}} \neq 0. \tag{40}$$

We can furthermore fix the phase of $|\phi_w\rangle_{\text{anc}}$ such that the overlap in (40) is real and positive. We can choose all other $|\phi_r\rangle_{\text{anc}}$ for $r \neq w, s$ arbitrary with the condition that if the overlap $\langle \tilde{v}_s|_{\text{anc}} \otimes \langle \tilde{s}|_{\text{reg}} U |\phi_r\rangle_{\text{anc}} \otimes |r\rangle_{\text{reg}}$ is non-zero, we adjust the phase so that the overlap is real and positive. From this it follows

$$\mu \langle \tilde{v}_s|\tilde{\eta}\rangle_{\text{anc}} = \langle \tilde{v}_s|_{\text{anc}} \otimes \langle \tilde{s}|_{\text{reg}} U \sum_{\substack{r=1 \\ r \neq s}}^{N_r} |\phi_r\rangle_{\text{anc}} \otimes |r\rangle_{\text{reg}} \in \mathbb{R}_{>0} \tag{41}$$

which concludes claim 1.
To prove the more restrictive statement (*ii*), it remains to show:
  a) $|r\rangle_{\text{reg}} \mapsto |\tilde{r}\rangle_{\text{reg}}$ is a well-defined bijective function, $\tilde{r} = f(r)$,
       $f : \{1, \dots, N_r\} \to \{1, \dots, N_r\}$
  b) $|\phi_r\rangle_{\text{anc}} \mapsto |\tilde{v}_r\rangle_{\text{anc}} = \mathbb{1} \otimes \langle f(r)| U |\phi_r\rangle_{\text{anc}} |r\rangle_{\text{reg}}$ is unitary $\forall |\phi_r\rangle_{\text{anc}} \in \mathcal{H}_{\text{anc}}, 1 \leq r \leq N_r$
To show that f is a well-defined function, we need to show in addition to claim 1 that the registers are mapped independently of the state of the ancilla. Assume this was not the case, i.e. $\exists r, |\phi_r^1\rangle_{\text{anc}}, |\phi_r^2\rangle_{\text{anc}}$ s.t.

$$U |\phi_r^i\rangle_{\text{anc}} \otimes |r\rangle_{\text{reg}} = |\tilde{v}_r^i\rangle_{\text{reg}} \otimes |\tilde{r}^i\rangle_{\text{reg}} \tag{42}$$

with $\tilde{r}^1 \neq \tilde{r}^2$. Consider two register-uniform states which are identical besides one containing $|\phi_r^1\rangle_{\mathrm{anc}} \otimes |r\rangle_{\mathrm{reg}}$ and the other $|\phi_r^2\rangle_{\mathrm{anc}} \otimes |r\rangle_{\mathrm{reg}}$. Then due to linearity, one of the two states is not mapped to a register-uniform state under U. This is a contradiction and hence f is a well-defined function. Because U is register-preserving, f must be surjective and is hence bijective.

To show b) we note that for any r the restriction $U : \mathrm{span}\{\mathcal{H}_{\mathrm{anc}} \otimes |r\rangle\} \to \mathrm{span}\{\mathcal{H}_{\mathrm{anc}} \otimes |f(r)\rangle\}$ is unitary.

$(ii) \implies (iii)$: Given U_r and $f(\circ)$ as in (ii), we can directly define a sequence of conditioned unitaries and a permutation on the registers.

For any r we can condition the unitary U_r acting on $\mathcal{H}_{\mathrm{anc}}$ on the bitvector-representation of r. Let us call the resulting unitary CU_r. Define the permutation $P : \mathcal{H}_{\mathrm{reg}} \ni |r\rangle \mapsto |f(r)\rangle \in \mathcal{H}_{\mathrm{reg}}$. Then

$$U = \mathbb{1}_{\mathrm{anc}} \otimes P \prod_{r=1}^{N_r} CU_r \tag{43}$$

$(iii) \implies (i)$: As the set of register-preserving unitaries is closed under composition, it suffices to show that basis-permutations on the registers and unitaries on the ancillas conditioned on register-states are both register-preserving. Both can be verified explicitly by tracking the action on a state of the form $\frac{1}{\sqrt{N_r}} \sum_{r=1}^{N_r} |\phi_r\rangle_{\mathrm{anc}} \otimes |r\rangle_{\mathrm{reg}}$ and observing that the normalization of the individual terms in the sum is not compromised. Permutations on the register merely reassign the ancilla states to different addresses while conditional unitaries on the ancillas apply unitary transformations on the ancilla-states for a subset of registers specified through the conditioning. $\qquad\square$

Lemma 2 *For register-uniform states as in equation* (23) *with* $\langle\phi_{r_1}|\phi_{r_2}\rangle_{anc} \in \mathbb{R}\ \forall r_1, r_2 \in \{1, \ldots, N_r\}$, *a unitary* $U = \mathbb{1}_{anc} \otimes U_{reg}$ *only non-trivially acting on the register-qubits always maps* $|\psi\rangle$ *to a register-uniform state if and only if*

$$(U^{\dagger})_{r_1 s} U_{s r_2} \in i\mathbb{R}\ \forall r_1, r_2, s \in \{1, \ldots, N_r\}\ with\ r_1 < r_2. \tag{44}$$

Proof We will show the claim by demanding $U|\psi\rangle$ adhere to definition 1.

Projecting $U|\psi\rangle$ on a register s, we obtain

$$\mathbb{1}_{\mathrm{anc}} \otimes \langle s|_{\mathrm{reg}} \mathbb{1} \otimes U_{\mathrm{reg}} \frac{1}{\sqrt{N_r}} \sum_{r=1}^{N_r} |\phi_r\rangle_{\mathrm{anc}} \otimes |r\rangle_{\mathrm{reg}} = \frac{1}{\sqrt{N_r}} |\tilde{\phi}_s\rangle_{\mathrm{anc}} \tag{45}$$

and therefore

$$\langle\tilde{\phi}_s|\tilde{\phi}_s\rangle_{\mathrm{anc}} = 1 \iff \sum_{r_1, r_2 = 1}^{N_r} \langle\phi_{r_1}|\phi_{r_2}\rangle_{\mathrm{anc}} \langle r_1| U_{\mathrm{reg}}^{\dagger} |s\rangle_{\mathrm{reg}} \langle s| U_{\mathrm{reg}} |r_2\rangle_{\mathrm{reg}} = 1 \tag{46}$$

We can split the sum to obtain (using the assumption of real-valued ancilla overlap)

$$1 = \underbrace{\sum_{r=1}^{N_r} \langle r| U_{\mathrm{reg}}^{\dagger} |s\rangle_{\mathrm{reg}} \langle s| U_{\mathrm{reg}} |r\rangle_{\mathrm{reg}}}_{=1}$$

$$+ \sum_{1 \leq r_1 < r_2 \leq N_r} \langle \phi_{r_1} | \phi_{r_2} \rangle_{\text{anc}}$$

$$\times \underbrace{\left[\langle r_1 | U_{\text{reg}}^\dagger | s \rangle_{\text{reg}} \langle s | U_{\text{reg}} | r_2 \rangle_{\text{reg}} + \langle r_2 | U_{\text{reg}}^\dagger | s \rangle_{\text{reg}} \langle s | U_{\text{reg}} | r_1 \rangle_{\text{reg}} \right]}_{= 2 \, \text{Re} \left[(U^\dagger)_{r_1 s} U_{s r_2} \right]} \quad (47)$$

where the first term equals 1 due to $U_{\text{reg}} | r \rangle_{\text{reg}}$ forming an orthonormal basis of $\mathcal{H}_{\text{reg}}$. As this equations has to hold for all sets $\{ | \phi_r \rangle_{\text{anc}} \}_{r=1}^{N_r}$ which have real-valued overlap (hence we can engineer them such that exactly one chosen term $\langle \phi_{r_1} | \phi_{r_2} \rangle_{\text{anc}}$ in the sum is non-zero) and for all s, equation (47) is equivalent to $\text{Re} \left[(U^\dagger)_{r_1 s} U_{s r_2} \right] = 0 \; \forall r_1, r_2, s \in \{ 1, \ldots, N_r \}$ with $r_1 < r_2$. $\qquad \square$

The resulting state is register-uniform due to cancellation of phases. While this only holds for real-valued inner product between ancilla-states, it raises the question whether similar results are possible more generally if a record of the phases of the ancilla states is kept. Furthermore, we did not explore the possibility of allowing different register bases for the input and output state in the definition of a register-preserving unitary.

Transaction settlement problem   For some combinatorial graph problems such as Max-Cut, the corresponding QUBO matrix Q is in simple correspondence with the graph and its adjacency matrix. This is not the case here, where deciding which transactions to settle corresponds to choosing a subset of *edges* (not nodes, cf. Fig. 1). Quadratic terms beyond the graph connectivity are common for QUBO problems which incorporate constraints as quadratic penalties. This increase the number of non-zero off-diagonal elements of Q and hence limits the applicability of many NISQ-QS as stressed in Sect. 1.1. For transaction settlements, we can relate the number of non-zero elements per row of Q as follows:

Lemma 3 *Given a transaction settlement with $A, \underline{\mathbf{b}}(\underline{\mathbf{s}})$ as in equations (7) and (8), represented by a graph G with parties as K nodes $\mathcal{K}$ and transactions as I edges $\mathcal{I}$ connecting the transacting parties, then: The average number of non-zero entries per row of the matrix $Q = A + Diag(\underline{\mathbf{b}}(\underline{\mathbf{s}}))$ is bounded by*

$$\mathbb{E}_{\mathcal{I}} \left[\sum_{j=1}^{I} \delta_{\{ Q_{ij} \neq 0 \}} \right] \leq 4 \frac{I}{K} + \frac{K}{I} \mathbb{V}_{\mathcal{K}} [N_k] - 1 \quad (48)$$

where N_k is the number of edges connected to node $k \in \mathcal{K}$.

For *d-regular graphs* $\mathbb{V}_{\mathcal{K}} [N_k] = 0$.

Proof We are looking for an upper-bound of the average number of non-zero elements in the rows of Q. For this assume the diagonal elements are all non-zero. For the off-diagonal elements we only need to consider the contributions of $A = -\lambda V V^T$. As $V_{il} := (\underline{\mathbf{v}}_{ik(l)})_{j(l)}$ is only non-zero if transaction i changes balance $j(l)$ of party $k(l)$, A_{ij} may only be non-zero if edges i and j share a node. Therefore

$$\mathbb{E}_{\mathcal{I}} \left[\sum_{j=1}^{I} \delta_{\{ Q_{ij} \neq 0 \}} \right] = \frac{1}{I} \sum_{i=1}^{I} \left[\sum_{j=1}^{I} \delta_{\{ Q_{ij} \neq 0 \}} \right]$$

$$\leq \frac{1}{I} \sum_{i=1}^{I} \left[\underbrace{\sum_{j \in \mathcal{I}} \delta_{\{\text{Transaction } i \& j \text{ share a node}\}}}_{=:a_i} \right]. \tag{49}$$

Let $k_1(i), k_2(i) \in \mathcal{K}$ such that $i = (k_1(i), k_2(i))$.[13] Then

$$a_i \leq N_{k_1(i)} + N_{k_2(i)} - 1$$

where -1 comes from double-counting the edge i connecting $k_1(i)$ with $k_2(i)$. Plugging this into equ. (49) we get

$$\mathbb{E}_{\mathcal{I}} \left[\sum_{j=1}^{I} \delta_{\{Q_{ij} \neq 0\}} \right] \leq -1 + \frac{1}{I} \sum_{k=1}^{K} N_k \left[\sum_{i \in \mathcal{I}} \delta_{\{i_0 = k\}} + \delta_{\{i_1 = k\}} \right] \tag{50}$$

$$= -1 + \frac{K}{I} \left[\frac{1}{K} \sum_{k=1}^{K} N_k^2 \right]$$

$$= -1 + \frac{K}{I} \left[\mathbb{E}_{\mathcal{K}}[N_k]^2 + \mathbb{E}_{\mathcal{K}}[N_k^2] - \mathbb{E}_{\mathcal{K}}[N_k]^2 \right] \tag{51}$$

$$= -1 + \frac{K}{I} \left[\left(2\frac{I}{K} \right)^2 + \mathbb{V}_{\mathcal{K}}[N_k] \right] \tag{52}$$

$$= 4\frac{I}{K} + \frac{K}{I} \mathbb{V}_{\mathcal{K}}[N_k] - 1 \tag{53}$$

$\square$

Appendix C: Cost estimator and marginal probability distribution obtained from sampling algorithm

In this section, we give explicit formulas for the estimators in Sect. 2.3 and their gradients. For the special case of a disjoint covering and greedy sampling algorithm, we rigorously prove the heuristic estimators used in the cost objective.

C.1 Cost estimator

To express $\hat{p}_i$ and $\hat{p}_{ij}$ through a set of measurements $\mathcal{M} = \left\{ \underline{\mathbf{m}}_1, \dots \underline{\mathbf{m}}_{n_{\text{shots}}} \right\}$ in the computational basis, let $\underline{\tilde{\mathbf{x}}}(\underline{\mathbf{m}})$ refer to the "bit"-vector $\in \{-1, 0, 1\}^I$ with the subset $A_{r(\underline{\mathbf{m}})}$ of entries fixed to $b(\underline{\mathbf{m}})_1 \dots b(\underline{\mathbf{m}})_{n_a}$ as described in Sect. 2.2 and all other bits set to -1. Define $l_r(\circ) : A_r \to \{1, \dots, n_a\}$ through $A_r[l_r(i)] = i \; \forall i \in A_r$ ("bit i is mapped to $l_r(i)^{\text{th}}$ ancilla bit of register r"). The estimators are given as follows:

$$\hat{p}_i(\underline{\theta}) := \frac{\sum_{\underline{\mathbf{m}} \in \mathcal{M}} \delta_{\tilde{x}_i(\underline{\mathbf{m}}), 1}}{\sum_{\underline{\mathbf{m}} \in \mathcal{M}} (1 - \delta_{\tilde{x}_i(\underline{\mathbf{m}}), -1})} \tag{54}$$

$$\hat{p}_{ij}(\underline{\theta}) := (1 - \hat{\mu}_{ij}) \hat{q}_{ij} + \hat{\mu}_{ij} \hat{p}_i \hat{p}_j \tag{55}$$

---

[13]In the case of an undirected graph we can fix any ordering.

where:

$$\hat{q}_{ij}(\underline{\theta}) := \frac{\sum_{\underline{m} \in \mathcal{M}} \delta_{\tilde{x}_i(\underline{m}),1} \delta_{\tilde{x}_j(\underline{m}),1}}{\sum_{\underline{m} \in \mathcal{M}} (1 - \delta_{\tilde{x}_i(\underline{m}),-1})(1 - \delta_{\tilde{x}_j(\underline{m}),-1})} \tag{56}$$

$$\hat{\mu}_{ij}(\underline{\theta})$$

$$:= \frac{\sqrt{\sum_{\underline{m} \in \mathcal{M}}(1 - \delta_{\tilde{x}_i(\underline{m}),-1}) \sum_{\underline{m} \in \mathcal{M}}(1 - \delta_{\tilde{x}_j(\underline{m}),-1})}}{\sqrt{\sum_{\underline{m} \in \mathcal{M}}(1 - \delta_{\tilde{x}_i(\underline{m}),-1}) \sum_{\underline{m} \in \mathcal{M}}(1 - \delta_{\tilde{x}_j(\underline{m}),-1})} + \sum_{\underline{m} \in \mathcal{M}}(1 - \delta_{\tilde{x}_i(\underline{m}),-1})(1 - \delta_{\tilde{x}_j(\underline{m}),-1})} \tag{57}$$

Note, that for disjoint coverings $\mathcal{A}$, the sums over r collapse to at most a single term as every bit i is contained in exactly one register-set $A_{r(i)}$. In the definition of $\tilde{q}_{ij}$ we adopt the convention $\tilde{q}_{ij} \equiv 0$ if $\{r : i, j \in A_r\} = \emptyset$ (sum contains no terms).

C.2 Derivation for disjoint covering

To derive the marginal probability distributions $p_{ij}(\underline{\theta}) = \text{Prob}_{\underline{\theta}}(x_i = 1, x_j = 1)$ and $p_i(\underline{\theta}) = \text{Prob}_{\underline{\theta}}(x_i = 1)$ that arise from the greedy sampling algorithm, consider the output of the PQC given by equation (16).

We denote the sequence of registers during one run of the sampling algorithm by $S = (r_i)_{i=1}^{N_m}$, where r_i is the register sampled in the ith measurement (omitting measurements with not bits being set)and N_m is the number of such calls before termination of the algorithm. Denote by $\tilde{A}_{r_i}^{(S)} \subset A_{r_i}$[14] the bits set in the ith measurement.

Then

$$\text{Prob}(\underline{x}) = \sum_S \text{Prob}(S) \prod_{i=1}^{N_m(S)} p_{\tilde{A}_i^{(S)}}^{(r_i)}(\underline{x}) \tag{58}$$

where $\text{Prob}(S)$ is the probability of the sampling algorithm resulting in the register-sequence S and $p_{\tilde{A}_{r_i}^{(S)}}^{(r_i)}(\underline{x})$ is defined through the complex ancilla amplitudes compatible with $\underline{x}_{\tilde{A}_i^{(S)}}$ for the given register r_i, i.e.

$$p_{\tilde{A}_i^{(S)}}^{(r_i)}(\underline{x}) = \sum_{b_k \begin{cases} \in \{0,1\}, & \text{for } A_{r_i}[j] \notin \tilde{A}_{r_i}^{(S)} \\ = x_{A_{r_i}[k]}, & \text{for } A_{r_i}[k] \in \tilde{A}_{r_i}^{(S)} \end{cases}} |a_r^{b_1 \dots b_{n_a}}|^2$$

which simplifies to $|a^{\underline{x}_{A_r}}|^2$ if $|\tilde{A}_{r_i}^{(S)}| = n_a$.

For fixed S define $\tilde{r}(i)$, $i \in B$ such that $i \in \tilde{A}_{\tilde{r}(i)}^{(S)}$, as well as $p_i^{(r)} := p_{\{i\}}^{(r)}(\underline{1}) = \sum_{\substack{b_k \in \{0,1\} \\ b_{l_r(i)}=1}} |a_r^{b_1 \dots b_{n_a}}|^2$ and $p_{ij}^{(r)} := p_{\{i,j\}}^{(r)}(\underline{1}) = \sum_{\substack{b_k \in \{0,1\} \\ b_{l_r(i)}=b_{l_r(j)}=1}} |a_r^{b_1 \dots b_{n_a}}|^2$. Then it follows for the marginals:

$$p_i = \sum_{\underline{x}: x_i=1} \text{Prob}(\underline{x}) = \sum_S \text{Prob}(S) p_i^{(\tilde{r}(i))} \tag{59}$$

---

[14]In the case of a disjoint covering $\mathcal{A}$, we have $\tilde{A}_{r_i}^{(S)} = A_{r_i}$.

$$p_{ij} = \sum_{\underline{x}:\ x_i=1, x_j=1} \text{Prob}(\underline{x}) = \sum_{S:\ \tilde{r}(i) \neq \tilde{r}(j)} \text{Prob}(S) p_i^{(\tilde{r}(i))} p_j^{(\tilde{r}(j))} + \sum_{S:\ \tilde{r}(i)=\tilde{r}(j)} \text{Prob}(S) p_{ij}^{(\tilde{r}(i))} \tag{60}$$

In the case of a disjoint covering $\mathcal{A}$, every register has to be sampled once and $\tilde{A}_{r_i}^{(S)} = A_{r_i}\ \forall i \in \{1, \ldots, N_m = \frac{I}{n_a}\}$. Due to this the sum over S merely consists of different orderings of $\{1, \ldots, N_r = \frac{I}{n_a}\}$. Therefore, $\text{Prob}(S) = \prod_{i=1}^{N_m} |\beta_{r_i}|^2 = (\frac{I}{n_a}!)^{-1}$ is uniform.

For general coverings however, $\text{Prob}(S)$ is not necessarily uniform and the sum over S is non-trivial. However, we can *hope to approximate* in the general case

$$p_i = \sum_{S} \text{Prob}(S) p_i^{(r(i))} \approx \frac{1}{\sum_{r:\ i \in A_r} |\beta_r|^2} \sum_{r:\ i \in A_r} |\beta_r|^2 p_i^{(r)} \tag{61}$$

and

$$p_{ij} = \sum_{S:\ r(i) \neq r(j)} \text{Prob}(S) p_i^{(r(i))} p_j^{(r(j))} + \sum_{S:\ r(i)=r(j)} \text{Prob}(S) p_{ij}^{(r(i))} \tag{62}$$

$$\approx \mu_{ij} \frac{1}{\sum_{r_1, r_2:\ i \in A_{r_1}, j \notin A_{r_1}, j \in A_{r_2}} |\beta_{r_1}|^2 |\beta_{r_2}|^2} \sum_{r_1, r_2:\ i \in A_{r_1}, j \notin A_{r_1}, j \in A_{r_2}} |\beta_{r_1}|^2 |\beta_{r_2}|^2 p_i^{(r_1)} p_j^{(r_2)} \tag{63}$$

$$+ (1 - \mu_{ij}) \frac{1}{\sum_{r:\ i,j \in A_r} |\beta_r|^2} \sum_{r:\ i,j \in A_r} |\beta_r|^2 p_{ij}^{(r)} \tag{64}$$

(equality holds for disjoint coverings, in which case the sums become trivial). We can regard this as assuming that based on symmetry considerations,[15] the probability of the sampling algorithm running through the sequence S where entry i is sampled from register $r \in \{r:\ i \in A_r\})$ is proportional to $|\beta_r|^2$. Similar considerations are made when looking at bit-pairs (i, j) with the added complexity that they can either be sampled from the same register r or from two different register r_1, r_2. The probability of the former is estimated as

$$\mu_{ij} \approx \frac{\sqrt{\sum_{r_1, r_2:\ i \in A_{r_1}, j \notin A_{r_1}, j \in A_{r_2}} |\beta_{r_1}|^2 |\beta_{r_2}|^2}}{\sqrt{\sum_{r_1, r_2:\ i \in A_{r_1}, j \notin A_{r_1}, j \in A_{r_2}} |\beta_{r_1}|^2 |\beta_{r_2}|^2} + \sum_{r:\ i,j \in A_r} |\beta_r|^2}. \tag{65}$$

By further approximating $\sum_{r_1, r_2:\ i \in A_{r_1}, j \notin A_{r_1}, j \in A_{r_2}} \approx \sum_{r_1:\ i \in A_{r_1}} \sum_{r_2:\ j \in A_{r_2}}$ we obtain

$$p_{ij} \approx \mu_{ij} \sum_{r_1:\ i \in A_{r_1}} \sum_{r_2:\ j \in A_{r_2}} p_i^{(r_1)} p_j^{(r_2)} + (1 - \mu_{ij}) \frac{1}{\sum_{r:\ i,j \in A_r} |\beta_r|^2} \sum_{r:\ i,j \in r} |\beta_r|^2 p_{ij}^{(r)}. \tag{66}$$

Given this, p_i is estimated by $\hat{p}_i$ (equ. (54)) and p_{ij} by $\hat{p}_{ij}$ (equ. (55)), where all "$\approx$" are exact for the case of a disjoint covering.

In summary, we have motivated the cost estimator in Sect. 2.3 as a heuristic for the general case and proven

---

[15] which depending on the graph covering $\mathcal{A}$ may not be warranted.

Lemma 4 *For the greedy sampling algorithmand a disjoint covering $\mathcal{A}$ (perfect matching), equation* (21) *and* $\mathbb{E}_\theta[C]$ *from equation* (15) *are equal in the limit* $n_{shots} \to \infty$. *In particular*

$$p_i(\underline{\theta}) = \sum_{\substack{b_k \in \{0,1\} \\ b_{l_r(i)}=1}} |a_r^{b_1 \dots b_{n_a}}|^2 \ where \ r \ s.t. \ i \in A_r \tag{67}$$

$$p_{ij}(\underline{\theta}) = \begin{cases} \sum_{\substack{b_k \in \{0,1\} \\ b_{l_{r_i}(i)}=b_{l_{r_i}(j)}=1}} |a_{r_i}^{b_1 \dots b_{n_a}}|^2, & if \ r_i = r_j \\ \sum_{\substack{b_k \in \{0,1\} \\ b_{l_{r_i}(i)}=1}} |a_{r_i}^{b_1 \dots b_{n_a}}|^2 \sum_{\substack{b_k \in \{0,1\} \\ b_{l_{r_j}(j)}=1}} |a_{r_j}^{b_1 \dots b_{n_a}}|^2, & if \ r_i \neq r_j \end{cases} \ where \ r_x \ s.t. \ x \in A_{r_x} \tag{68}$$

C.3 Explicit form of cost-gradient

In this section, we will give the explicit form of the derivatives ∂_{θ_d} of the cost a and register-regularization estimator

$$\hat{C}(\underline{\theta}) + \hat{R}(\underline{\theta})$$
$$= \sum_{\substack{i,j=1 \\ i \neq j}}^{I} \hat{p}_{ij}(\underline{\theta}) A_{ij} + \sum_{i=1}^{I} \hat{p}_i(\underline{\theta})(A_{ii} + b_i(\underline{\hat{s}}(\underline{\theta}))) + c(\underline{\hat{s}}(\underline{\theta})) + \eta \sum_{r=1}^{N_r} \left[\hat{r}_r(\underline{\theta}) - \frac{1}{N_r} \right]^2. \tag{69}$$

When optimizing $\hat{C}(\underline{\theta})$ with respect to $\underline{\theta}$ we may want to make use of the gradient $\nabla_{\underline{\theta}}\hat{C}$ to update $\underline{\theta}$. For this, note that both terms in the quotient for $\hat{p}_i$ and $\hat{q}_{ij}$ are given as linear combinations of Pauli-expectation values (equation (54) and (56)). For many variational ansätze, in particular those consisting of single-qubit Pauli-rotations used in this paper, this allows to easily calculate gradients $\nabla_{\underline{\theta}}[\hat{p}_i]$ and $\nabla_{\underline{\theta}}[\hat{q}_{ij}]$ through the parameter-shift rule ([92, 97]). Similarly, the gradients of $\hat{\mu}_{ij}$ and $\hat{r}_r$ can be calculated by applying the parameter-shift rule. Through application of the chain-rule, the gradient of $\hat{C}\left(\{\hat{p}_i\}_i, \{\hat{q}_{ij}\}_{ij}, \{\hat{\mu}_{ij}\}_{ij}; \{\hat{r}_r\}_r\right)$ is obtained.

We distinguish between the case of register-preserving ansätze and general circuit ansätze. For simplicity (and motivated by the regularization of the register-probabilities) we will treat $\hat{\mu}_{ij}$ as constant in $\underline{\theta}$ in both cases.

By the chain rule we get

$$\partial_{\theta_d}\hat{C}(\underline{\theta}) = \sum_{\substack{i,j=1 \\ i \neq j}}^{I} \partial_{\theta_d}\hat{p}_{ij} A_{ij} + \sum_{i=1}^{I} \left[\partial_{\theta_d}\hat{p}_i(A_{ii} + b_i(\underline{\hat{s}})) + \hat{p}_i \nabla_{\underline{s}} b_i^T(\underline{\hat{s}}) \partial_{\theta_d}\underline{\hat{s}} \right] + \nabla_{\underline{s}}c(\underline{\hat{s}}) \partial_{\theta_d}\underline{\hat{s}} \tag{70}$$

$$\partial_{\theta_d}\hat{R}(\underline{\theta}) = 2\eta \sum_{r=1}^{N_r} \left[\hat{r}_r - \frac{1}{N_r} \right] \partial_{\theta_d}\hat{r}_r \tag{71}$$

where

$$\nabla_{\underline{s}} b_i = 2\lambda V_{(i)} \tag{72}$$

$$\nabla_{\underline{s}} c = -2\lambda \left(\underline{l} + \underline{s} - \underline{\textbf{bal}} \right) \tag{73}$$

$$\partial_{\theta_d}\underline{\hat{s}} = \text{Diag}[(\delta_{\hat{s}_i>0})_{1 \leq i \leq KJ}] \sum_{i=1}^{I} \partial_{\theta_d}\hat{p}_i V_{(i)} \tag{74}$$

$$\partial_{\theta_d}\hat{p}_{ij} = (1 - \hat{\mu}_{ij})\partial_{\theta_d}\hat{q}_{ij} + \hat{\mu}_{ij}(\hat{p}_j\partial_{\theta_d}\hat{p}_i + \hat{p}_i\partial_{\theta_d}\hat{p}_j) \tag{75}$$

with $V_{(i)}$ being the ith row of V. As $\hat{r}_r$ is the expectation of the observable $\mathbb{1}_{n_a} \otimes |r\rangle \langle r|$ its partial derivatives can be calculated directly through parameter-shift rules or similar techniques to calculate the gradient of quantum observables. If we consider a register-preserving ansatz, the same holds true for $\hat{q}_{ij}(\underline{\theta})$ and $\hat{p}_i(\underline{\theta})$. Otherwise, the derivatives have to be calculated separately for the nominator and denominator in equation (54) and (56) respectively and recombined using the quotient rule.

Appendix D: Expressing cost estimator as Hermitian observable

We observed in Sect. 2.2 and 2.3, that our qubit-compression results in a cost-estimator expressed as a function of Pauli-Z measurements that cannot be written straightforwardly as the expectation over a Hermitian operator. Here, we alleviate this issue for register-preserving circuits and fixed slack variables.

We can rewrite the cost as a Hermitian expectation for register-preserving circuits and fixed slack variables **s**:

1. Substitute denominators in $\hat{p}_i$ and $\hat{q}_{ij}$ as well as $\hat{\mu}_{ij}$ by exact counterpart.
2. Define operator $C(\underline{\theta})$ such that $\hat{C}(\underline{\theta}) = \langle C(\underline{\theta})\rangle_{\mathcal{M}}$ by "doubling the Hilbert space".

1   The denominators in the expressions for $\hat{p}_i$ and $\hat{q}_{ij}$ (equations (54) and (56)) as well as $\hat{\mu}_{ij}$ can be replaced by scalar constants by substituting

$$\langle \mathbb{1}_{n_a} \otimes |r\rangle \langle r|\rangle_{\mathcal{M}} \mapsto \frac{1}{N_r} \tag{76}$$

resulting in ($n_i := \sum_{\substack{r=1 \\ i\in A_r}}^{N_r} 1$, $n_{ij} := \sum_{\substack{r=1 \\ i,j\in A_r}}^{N_r} 1$)

$$\hat{\mu}_{ij} = \frac{\sqrt{n_i n_j}}{\sqrt{n_i n_j} + n_{ij}} \tag{77}$$

$$\hat{p}_i = \frac{N_r}{n_i}\left\langle \sum_{\substack{r=1 \\ i\in A_r}}^{N_r} \mathbb{1}_{l_r(i)-1} \otimes |1\rangle \langle 1| \otimes \mathbb{1}_{n_a-l_r(i)} \otimes |r\rangle \langle r|\right\rangle_{\mathcal{M}} \tag{78}$$

$$\hat{q}_{ij} = \frac{N_r}{n_{ij}}\left\langle \sum_{\substack{r=1 \\ i,j\in A_r}}^{N_r} \mathbb{1}_{\min\{l_r(i),l_r(j)\}-1} \otimes |1\rangle \langle 1| \otimes \mathbb{1}_{|l_r(i)-l_r(j)|-1} \otimes |1\rangle \right.$$

$$\left. \times \langle 1| \otimes \mathbb{1}_{n_a-l_r(i)-l_r(j)} \otimes |r\rangle \langle r|\right\rangle_{\mathcal{M}} \tag{79}$$

where we set $\hat{p}_i = 0$ ($\hat{q}_{ij} = 0$) if $n_i = 0$ ($n_{ij} = 0$).

2   Equations (78) and (79) suggest defining the Hermitian operators

$$P_i = \frac{N_r}{n_i}\sum_{\substack{r=1 \\ i\in A_r}}^{N_r} \mathbb{1}_{l_r(i)-1} \otimes |1\rangle \langle 1| \otimes \mathbb{1}_{n_a-l_r(i)} \otimes |r\rangle \langle r| \tag{80}$$

$$Q_{ij} = \frac{N_r}{n_{ij}} \sum_{\substack{r=1 \\ i,j \in A_r}}^{N_r} \mathbb{1}_{\min\{l_r(i),l_r(j)\}-1} \otimes |1\rangle$$

$$\times \langle 1| \otimes \mathbb{1}_{|l_r(i)-l_r(j)|-1} \otimes |1\rangle \langle 1| \otimes \mathbb{1}_{n_a-l_r(i)-l_r(j)} \otimes |r\rangle \langle r|. \tag{81}$$

This would allow us to write $\hat{C}(\underline{\theta})$ in the desired form if it weren't for the terms of the form $\hat{p}_i \hat{p}_j = \langle P_i \rangle_{\mathcal{M}} \langle P_j \rangle_{\mathcal{M}}$. Instead of a quantum state $|\psi(\underline{\theta})\rangle$ in the form of equation (16), we consider the product state $|\psi(\underline{\theta})\rangle \otimes |\psi(\underline{\theta})\rangle$ and the following operator acting on it:

$$C = . \sum_{i,j=1}^{I} A_{ij} \left[(1 - \hat{\mu}_{ij}) Q_{ij} \otimes \mathbb{1}_{n_q} + \hat{\mu}_{ij} P_i \otimes P_j \right] + \sum_{i=1}^{I} (A_{ii} + b_i(\underline{s})) P_i \otimes \mathbb{1}_{n_q} + c(\underline{s}) \mathbb{1}_{2n_q} \tag{82}$$

This requires doubling the number of qubits and circuit width. The cost estimator is then given as this operator's expectation value

$$\hat{C}(\underline{\theta}) = \langle C \rangle_{\mathcal{M}} \simeq \langle \psi(\underline{\theta})| \otimes \langle \psi(\underline{\theta})| \, C \, |\psi(\underline{\theta})\rangle \otimes |\psi(\underline{\theta})\rangle \tag{83}$$

estimated through measurements.

We note:

- C is Hermitian as it is real-valued and diagonal in the computational basis
- While we assumed the measured quantum state to be register-uniform in the substitutions of 1., one may consider using the observable in equation (82) even for non-register-preserving circuits if the register-amplitudes are approximately kept constant with a penalty term. The penalty term can be added to C in the same manner as $P_i \otimes P_j$.
- The assumption of a register-preserving circuit comes with a caveat: We need to be careful in applying methods tailored to Hermitian-expectation-minimization which change the variational ansatz itself based on properties of the Hermitian (QAOA being the most prominent example). Still there are relevant results which can be applied to a constrained set of allowed gates, such as optimization techniques ([92, 93]), estimation ([82]) and error mitigation techniques as well as libraries ([99–101]) or fault-tolerant methods for evaluating expectation values (e.g. [102]).
- Throughout, we assumed constant slack variables $\underline{s}$, as expressing the relationship in (22) through an observable is complicated by the non-linearity of the rectified linear unit. Hence, alternating adjustments of $\underline{s}$ using (22) and the circuit parameters $\underline{\theta}$ through a classical optimizer are needed. Alternatively, both variables could be optimized simultaneously, for which implicit differentiation may be useful ([103]).

Appendix E: Simulation parameters

The configurations used to determine ansatz and for optimizing circuit parameters in Sect. 3 are shown in Table 4. Pennylane ([98]; version: `pennylane=0.29.1`) was used for quantum computing simulations. For parameter optimization, the SciPy ([96]; version: `scipy=1.10.1`) implementation of gradient-free optimizer COBYLA as well as standard gradient descent were used. The gradients of the latter were calculated through the chain rule and parameter-shift rule ([92, 97]).

Table 4 Benchmarking setup

Parameter	Value(s)
Cost penalty λ	10
Register-regularization penalty* η	1000
Gradient max steps	1500
Gradient stepsize	2.5×10^{-4}
Optimizer	COBYLA, Gradient descent (DESC)
Number of runs / starting point	25
Parameter initialization	Uniform random $\in [0, 2\pi]$
Depth d	1, 4
Ancilla qubits n_a	1, 4, 8, 16
n_{shots}	$10^4, 2 \times 10^4$
Ansatz	Register-preserving, Hardware-efficient
Register-mapping	Disjoint covering unless $n_r + \geq 1$

* For Hardware-efficient ansatz only

Data Availability
Given permission by the regulated stock exchange, anonymized transaction data samples used and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
The authors consent to publication by EPJ Quantum Technology.

Competing interests
The authors declare no competing interests.

Author details
[1]D-CHAB Interdisciplinary Sciences, ETH Zürich, Vladimir-Prelog-Weg 1-5/10, 8093, Zurich, Switzerland. [2]Centre for Quantum Technologies, National University of Singapore, 3 Science Drive 2, Singapore, 117543, Singapore. [3]AngelQ Quantum Computing, 531A Upper Cross Street, 04-95 Hong Lim Complex, Singapore, 051531, Singapore. [4]School of Computing and Information Systems, Singapore Management University, 80 Stamford Road, Singapore, 178902, Singapore. [5]School of Electrical and Computer Engineering, Technical University of Crete, Chania, 73100, Greece.

References
1. Grover LK. A fast quantum mechanical algorithm for database search. In: Proceedings of the twenty-eighth annual ACM symposium on theory of computing. STOC'96. New York: Association for Computing Machinery; 1996. p. 212–9. https://doi.org/10.1145/237814.237866.
2. Shor PW. Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th annual symposium on foundations of computer science. 1994. p. 124–34. https://doi.org/10.1109/SFCS.1994.365700.
3. Harrow AW, Montanaro A. Quantum Computational Supremacy. arXiv:1809.07442v1 (2018).
4. Arute F, et al. Quantum supremacy using a programmable superconducting processor. Nature. 2019;574:505–10. https://doi.org/10.1038/s41586-019-1666-5.
5. Zhong H-S, et al. Quantum computational advantage using photons. Science. 2020;370:1460–3. https://doi.org/10.1126/science.abe8770.

6. Madsen LS, et al. Quantum computational advantage with a programmable photonic processor. Nature. 2022;606:75–81. https://doi.org/10.1038/s41586-022-04725-x.

7. Pan F, Chen K, Zhang P. Solving the sampling problem of the sycamore quantum circuits. Phys Rev Lett. 2022;129:090502. https://doi.org/10.1103/PhysRevLett.129.090502.

8. Kim Y, et al. Evidence for the utility of quantum computing before fault tolerance. Nature. 2023;618:500–5. https://doi.org/10.1038/s41586-023-06096-3.

9. Tindall J, et al. Efficient tensor network simulation of IBM's kicked Ising experiment. arXiv:2306.14887 (2023).

10. Kechedzhi K, et al. Effective quantum volume, fidelity and computational cost of noisy quantum processing experiments. arXiv:2306.15970 (2023).

11. Begušić T, Chan GK-L. Fast classical simulation of evidence for the utility of quantum computing before fault tolerance. arXiv:2306.16372 (2023).

12. Bharti K, et al. Noisy intermediate-scale quantum algorithms. Rev Mod Phys. 2022;94:015004. https://doi.org/10.1103/REVMODPHYS.94.015004/FIGURES/7/MEDIUM.

13. Wei J, et al. NISQ computing: where are we and where do we go? https://doi.org/10.1007/s43673-022-00058-z.

14. Tan B, et al. Qubit-efficient encoding schemes for binary optimisation problems. Quantum. 2021;5:454. https://doi.org/10.22331/q-2021-05-04-454.

15. Barahona F. On the computational complexity of Ising spin glass models. J Phys A, Math Gen. 1982;15:3241–53. https://doi.org/10.1088/0305-4470/15/10/028.

16. Date P, Arthur D, Pusey-Nazzaro L. QUBO formulations for training machine learning models. Sci Rep. 2021;11:10029. https://doi.org/10.1038/s41598-021-89461-4.

17. Vikstål P, et al. Applying the quantum approximate optimization algorithm to the tail assignment problem. Phys Rev Appl. 2020;14:034009. https://doi.org/10.1103/PhysRevApplied.14.034009.

18. Harwood S, et al. Formulating and solving routing problems on quantum computers. In: IEEE transactions on quantum engineering. vol. 2. 2021. p. 1–17. https://doi.org/10.1109/TQE.2021.3049230.

19. Braine L, et al. Quantum Algorithms for Mixed Binary Optimization applied to Transaction Settlement. arXiv:1910.05788 (2019). https://doi.org/10.1109/TQE.2021.3063635.

20. Punnen AP, editor. The quadratic unconstrained binary optimization problem: theory, algorithms, and applications. Cham: Springer; 2022. https://doi.org/10.1007/978-3-031-04520-2.

21. Finnila AB, et al. Quantum annealing: a new method for minimizing multidimensional functions. Chem Phys Lett. 1994;219:343–8. https://doi.org/10.1016/0009-2614(94)00117-0.

22. Kirkpatrick S. Optimization by simulated annealing: quantitative studies. J Stat Phys. 1984;34:975–86. https://doi.org/10.1007/BF01009452.

23. Yarkoni S, et al. Quantum annealing for industry applications: introduction and review. Rep Prog Phys. 2022;85:104001 https://doi.org/10.1088/1361-6633/ac8c54.

24. D.-Wave Systems Inc. D-wave hybrid solver service + advantage: technology update. Tech. Rep. https://www.dwavesys.com/media/m2xbmlhs/14-1048a-a_d-wave_hybrid_solver_service_plus_advantage_technology_update.pdf.

25. Farhi E, Goldstone J, Gutmann S. A Quantum Approximate Optimization Algorithm. arXiv:1411.4028 (2014).

26. Peruzzo A, et al. A variational eigenvalue solver on a photonic quantum processor. Nat Commun. 2014;5:4213. https://doi.org/10.1038/ncomms5213.

27. Tilly J, et al. The variational quantum eigensolver: a review of methods and best practices. Phys Rep. 2022;986:1–128. https://doi.org/10.1016/j.physrep.2022.08.003.

28. Benedetti M, et al. Parameterized quantum circuits as machine learning models. Quantum Sci Technol. 2019;4:043001 https://doi.org/10.1088/2058-9565/ab4eb5.

29. McClean JR, et al. The theory of variational hybrid quantum-classical algorithms. New J Phys. 2016;18:023023 https://doi.org/10.1088/1367-2630/18/2/023023.

30. McClean JR, et al. Barren plateaus in quantum neural network training landscapes. Nat Commun. 2018;9:4812 https://doi.org/10.1038/s41467-018-07090-4.

31. Arrasmith A, et al. Effect of barren plateaus on gradient-free optimization. Quantum. 2021;5:558 https://doi.org/10.22331/q-2021-10-05-558.

32. Wang S, et al. Noise-induced barren plateaus in variational quantum algorithms. Nat Commun. 2021;12:6961. https://doi.org/10.1038/s41467-021-27045-6

33. Liu X, et al. Mitigating barren plateaus of variational quantum eigensolvers. arXiv:2205.13539 (2022).

34. Pesah A, et al. Absence of barren plateaus in quantum convolutional neural networks. Phys Rev X. 2021;11:041011. https://doi.org/10.1103/PhysRevX.11.041011.

35. Patti TL, et al. Entanglement devised barren Plateau mitigation. Phys Rev Res. 2021;3:033090. https://doi.org/10.1103/PhysRevResearch.3.033090.

36. Grant E, et al. An initialization strategy for addressing barren plateaus in parametrized quantum circuits. Quantum. 2019;3:214. https://doi.org/10.22331/q-2019-12-09-214

37. Dborin J, et al. Matrix Product State Pre-Training for Quantum Machine Learning. arXiv:2106.05742 (2021).

38. Skolik A, et al. Layerwise learning for quantum neural networks. Quantum Mach Intell. 2021;3:5. https://doi.org/10.1007/s42484-020-00036-4.

39. Cerezo M, et al. Cost function dependent barren plateaus in shallow parametrized quantum circuits. Nat Commun. 2021;12:1791. https://doi.org/10.1038/s41467-021-21728-w

40. Schatzki L, et al. Theoretical Guarantees for Permutation-Equivariant Quantum Neural Networks. arXiv:2210.09974 (2022).

41. Sack SH, et al. Avoiding barren plateaus using classical shadows. PRX Quantum. 2022;3:020365. https://doi.org/10.1103/PRXQuantum.3.020365.

42. Seki K, Yunoki S. Quantum power method by a superposition of time-evolved states. PRX Quantum. 2021;2:010333. https://doi.org/10.1103/PRXQuantum.2.010333.

43. Kyriienko O. Quantum inverse iteration algorithm for programmable quantum simulators. npj Quantum Inf. 2020;6:1–8. https://doi.org/10.1038/s41534-019-0239-7.

44. Bharti K, Haug T. Iterative quantum-assisted eigensolver. Phys Rev A. 2021;104:L050401. https://doi.org/10.1103/PhysRevA.104.L050401.
45. Takeshita T, et al. Increasing the representation accuracy of quantum simulations of chemistry without extra quantum resources. Phys Rev X. 2020;10:011004. https://doi.org/10.1103/PhysRevX.10.011004.
46. Motta M, et al. Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution. Nat Phys. 2020;16:205–10. https://doi.org/10.1038/s41567-019-0704-4.
47. Huggins WJ, et al. A non-orthogonal variational quantum eigensolver. New J Phys. 2020;22:073009. https://doi.org/10.1088/1367-2630/ab867b.
48. Stair NH, Huang R, Evangelista FA. A multireference quantum Krylov algorithm for strongly correlated electrons. J Chem Theory Comput. 2020;16:2236–45. https://doi.org/10.1021/acs.jctc.9b01125.
49. Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual. https://www.gurobi.com (2023).
50. IBM ILOG Cplex. V12. 1: user's manual for CPLEX. In: International business machines corporation. vol. 46. 2009. p. 157.
51. Achterberg T. SCIP: solving constraint integer programs. Math Program Comput. 2009;1:1–41. https://doi.org/10.1007/s12532-008-0001-1.
52. Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. Science. 1983;220:671–80. https://doi.org/10.1126/science.220.4598.671.
53. Wang Y, et al. A multilevel algorithm for large unconstrained binary quadratic optimization. In: Beldiceanu N, Jussien N, Pinson É, editors. Integration of AI and OR techniques in contraint programming for combinatorial optimzation problems. Lecture notes in computer science. Berlin: Springer; 2012. p. 395–408. https://doi.org/10.1007/978-3-642-29828-8_26.
54. Goemans MX, Williamson DP. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. J ACM. 1995;42:1115–45. https://doi.org/10.1145/227683.227684.
55. Khot S, Kindler G, Mossel E. Optimal Inapproximability Results for MAX-CUT and Other 2-Variable CSPs? 2005.
56. Bennett CH, et al. Strengths and weaknesses of quantum computing. SIAM J Comput. 1997;26:1510–23. https://doi.org/10.1137/S0097539796300933.
57. Aaronson S. The limits of quantum. Sci Am. 2008;298:62–9.
58. Guerreschi GG, Matsuura AY. QAOA for max-cut requires hundreds of qubits for quantum speed-up. Sci Rep. 2019;9:6903. https://doi.org/10.1038/s41598-019-43176-9.
59. Date P, et al. Efficiently embedding QUBO problems on adiabatic quantum computers. Quantum Inf Process. 2019;18:117. https://doi.org/10.1007/s11128-019-2236-3.
60. Mitarai K, Fujii K. Overhead for simulating a non-local channel with local channels by quasiprobability sampling. Quantum. 2021;5:388. https://doi.org/10.22331/q-2021-01-28-388
61. Harrigan MP, et al. Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. Nat Phys. 2021;17:332–6. https://doi.org/10.1038/s41567-020-01105-y
62. Otterbach JS, et al. Unsupervised Machine Learning on a Hybrid Quantum Computer. arXiv:1712.05771 (2017).
63. Pelofske E, Bärtschi A, Eidenbenz S. Quantum Annealing vs. QAOA: 127 Qubit Higher-Order Ising Problems on NISQ Computers. arXiv:2301.00520v1.
64. Zhu Y, et al. Multi-round QAOA and advanced mixers on a trapped-ion quantum computer. Quantum Sci Technol. 2022;8:015007. https://doi.org/10.1088/2058-9565/ac91ef.
65. Shaydulin R, Pistoia M. QAOA with $N \cdot p \geq 200$. arXiv:2303.02064 (2023).
66. Dunjko V, Ge Y, Cirac Jl. Computational speedups using small quantum devices. Phys Rev Lett. 2018;121:25. https://doi.org/10.1103/PhysRevLett.121.250501.
67. Glover F, Lewis M, Kochenberger G. Logical and inequality implications for reducing the size and difficulty of quadratic unconstrained binary optimization problems. Eur J Oper Res. 2018;265:829–42. https://doi.org/10.1016/j.ejor.2017.08.025.
68. Lewis M, Glover F. Quadratic unconstrained binary optimization problem preprocessing: theory and empirical analysis. Networks. 2017;70:79–97. https://doi.org/10.1002/net.21751.
69. Fujii K, et al. Deep Variational Quantum Eigensolver: a divide-and-conquer method for solving a larger problem with smaller size quantum computers. arXiv:2007.10917 (2022).
70. Bechtold M, et al. Investigating the effect of circuit cutting in QAOA for the MaxCut problem on NISQ devices. arXiv:2302.01792 (2023).
71. Peng T, et al. Simulating large quantum circuits on a small quantum computer. Phys Rev Lett. 2020;125:150504. https://doi.org/10.1103/PhysRevLett.125.150504.
72. Amaro D, et al. Filtering variational quantum algorithms for combinatorial optimization. Quantum Sci Technol. 2022;7:015021. https://doi.org/10.1088/2058-9565/ac3e54.
73. Shee Y, et al. Qubit-efficient encoding scheme for quantum simulations of electronic structure. Phys Rev Res. 2022;4(2):023154. https://doi.org/10.1103/PhysRevResearch.4.023154.
74. Glos A, Krawiec A, Zimboras Z. Space-efficient binary optimization for variational quantum computing. npj Quantum Inf. 2022;8:39. https://doi.org/10.1038/s41534-022-00546-y.
75. Fuchs FG, et al. Efficient encoding of the weighted MAX *k*-CUT on a quantum computer using QAOA. SN Comput Sci. 2021;2:89. https://doi.org/10.1007/s42979-020-00437-z.
76. Fuller B, et al. Approximate Solutions of Combinatorial Problems via Quantum Relaxations. arXiv:2111.03167 (2021).
77. Teramoto K, et al. Quantum-Relaxation Based Optimization Algorithms: Theoretical Extensions. arXiv:2302.09481 (2023).
78. Rancic MJ. Noisy intermediate-scale quantum computing algorithm for solving an n-vertex MaxCut problem with log(n) qubits. Phys Rev Res. 2023;5(1):L012021. https://doi.org/10.1103/PhysRevResearch.5.L012021.
79. Winderl D, Franco N, Lorenz JM. A Comparative Study on Solving Optimization Problems with Exponentially Fewer Qubits. arXiv:2210.11823 (2022).
80. Guo Liu J, et al. Variational quantum eigensolver with fewer qubits. Phys Rev Res. 2019;1:023025. https://doi.org/10.1103/PhysRevResearch.1.023025.
81. Tibor Veszeli M, Vattay G. Mean Field Approximation for solving QUBO problems. arXiv:2106.03238 (2021).
82. Huang H-Y, Kueng R, Preskill J. Predicting many properties of a quantum system from very few measurements. Nat Phys. 2020;16:1050–7. https://doi.org/10.1038/s41567-020-0932-7.

83. Kuang R, Barbeau M. Quantum permutation pad for universal quantum-safe cryptography. Quantum Inf Process. 2022;21:21. https://doi.org/10.1007/s11128-022-03557-y.
84. Draper TG. Addition on a Quantum Computer. Tech. Rep. http://xxx.lanl.gov/quant-ph (2000).
85. Boneh A, Hofri M. The Coupon-Collector Problem Revisited (1989).
86. Truger F, et al. Warm-starting and quantum computing: a systematic mapping study. 2023. arXiv:2303.06133 [quant-ph].
87. Egger DJ, Mareček J, Woerner S. Warm-starting quantum optimization. Quantum. 2021;5:479 https://doi.org/10.22331/q-2021-06-17-479.
88. Akshay V, et al. Parameter concentrations in quantum approximate optimization. Phys Rev A. 2021;104:L010401. https://doi.org/10.1103/PhysRevA.104.L010401.
89. Mitarai K, Yan T, Fujii K. Generalization of the output of a variational quantum eigensolver by parameter interpolation with a low-depth ansatz. Phys Rev Appl. 2019;11:044087. https://doi.org/10.1103/PhysRevApplied.11.044087.
90. Thye Goh S, et al. Techniques to enhance a QUBO solver for permutation-based combinatorial optimization. In: GECCO 2022 companion - proceedings of the 2022 genetic and evolutionary computation conference. Association for Computing Machinery, Inc; 2022. p. 2223–31. https://doi.org/10.1145/3520304.3533982.
91. Kl Barkoutsos P, et al. Improving variational quantum optimization using CVaR. Quantum. 2020;4:256. https://doi.org/10.22331/q-2020-04-20-256.
92. Schuld M, et al. Evaluating analytic gradients on quantum hardware. Phys Rev A. 2019;99:032331. https://doi.org/10.1103/PhysRevA.99.032331.
93. Ostaszewski M, Grant E, Benedetti M. Structure optimization for parameterized quantum circuits. Quantum. 2021;5:391. https://doi.org/10.22331/q-2021-01-28-391.
94. Nakanishi KM, Fujii K, Todo S. Sequential minimal optimization for quantum-classical hybrid algorithms. Phys Rev Res. 2020;2:043158. https://doi.org/10.1103/PhysRevResearch.2.043158.
95. Bittel L, Kliesch M. Training variational quantum algorithms is NP-hard. Phys Rev Lett. 2021;127:120502. https://doi.org/10.1103/PhysRevLett.127.120502.
96. Virtanen P, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat Methods. 2020;17:261–72. https://doi.org/10.1038/s41592-019-0686-2.
97. Mitarai K, et al. Quantum circuit learning. Phys Rev A. 2018;98:032309. https://doi.org/10.1103/PhysRevA.98.032309.
98. Bergholm V, et al. PennyLane: Automatic differentiation of hybrid quantum-classical computations. arXiv:1811.04968 (2022).
99. O'Brien TE, et al. Error mitigation via verified phase estimation. PRX Quantum. 2021;2:020317. https://doi.org/10.1103/PRXQuantum.2.020317.
100.   Endo S, Benjamin SC, Li Y. Practical quantum error mitigation for near-future applications. Phys Rev X. 2018;8:031027. https://doi.org/10.1103/PhysRevX.8.031027.
101.   LaRose R, et al. Mitiq: a software package for error mitigation on noisy quantum computers. Quantum. 2022;6:774. https://doi.org/10.22331/q-2022-08-11-774
102.   Knill E, Ortiz G, Somma RD. Optimal quantum measurements of expectation values of observables. Phys Rev A. 2007;75:012328. https://doi.org/10.1103/PhysRevA.75.012328.
103.   Ahmed S, Killoran N, Álvarez JFC. Implicit differentiation of variational quantum algorithms. arXiv:2211.13765 (2022).

Publisher's Note

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