Blind three-qubit exact Grover search on a nitrogen-vacancy-center platform

Cica Gustiani [1,*] and David P. DiVincenzo [1,2,3,†]

[1]*Institute for Quantum Information, RWTH Aachen University, D-52056 Aachen, Germany*

[2]*Peter Grünberg Institute, Theoretical Nanoelectronics, Forschungszentrum Jülich, D-52425 Jülich, Germany*

[3]*Jülich-Aachen Research Alliance (JARA), Fundamentals of Future Information Technologies, D-52425 Jülich, Germany*

(Received 25 February 2021; revised 25 October 2021; accepted 20 October 2021; published 14 December 2021)

Here we put into practice the concept of blind client-server quantum computation, in which a client with limited quantum power controls the execution of a quantum computation on a powerful server, without revealing any details of the computation. In particular, it is a three-node setting in which an oracular quantum computation can be executed blindly. In this Blind Oracular Quantum Computation (BOQC), the oracle (Oscar) is another node, with limited power, who acts in cooperation with the client (Alice) to supply quantum information to the server so that the oracle part of the quantum computation can also be executed blindly. We develop tests of this protocol using two- and three-qubit versions of the exact Grover algorithm (i.e., with database sizes $4 \leqslant N \leqslant 8$), obtaining optimal implementations of these algorithms within a gate array scheme and the blinded cluster-state scheme. We discuss the feasibility of executing these protocols in state-of-the-art three-node experiments using nitrogen-vacancy diamond electronic and nuclear qubits.

DOI: 10.1103/PhysRevA.104.062422

I. INTRODUCTION

While the promise of distributed quantum information processing was already foreseen in theoretical work many decades ago [1–3], we have finally entered a time when some of these ideas can be implemented in the laboratory [4]. With these developments, it is timely to look at the theoretical situation in a new light and to evaluate what can be done with the currently very limited resources that are available. Some studies have already begun to assess the detailed resources needed to implement quantum algorithms [5,6], but these have not come to a point where any feasible, implementable schemes have been proposed.

In this paper we lay out a concrete plan for putting several different concepts in distributed quantum computing into action. Blind quantum computation [7] is an example of a protocol in which quantum physics gives unique security properties in a distributed computing setting. It is a client-server scheme, in which a client with limited computing power wishes to make use of a powerful server, but in such a way as to ensure that the server is "blind," i.e., not able to determine what computation the client is running, and not able to come into possession of any intelligible input or output data for this computation. It has been shown that an adaptation of the technique of cluster-state quantum computation [8–10] can achieve client-server blind quantum computation, and one aspect of our work in this paper will be to lay out the possibilities for achieving this in a distributed quantum device involving diamond nitrogen-vacancy (NV) centers.

It has been standard for 20 years to use oracular algorithms as test cases for quantum computing implementations. We will adopt this approach here as well, but we proposed in a detailed paper on the cryptographic protocol [11], a different approach to integrating oracular computations into the client-server paradigm. In a distributed setting, it is meaningful to consider the oracle to be a distinct node of a network. In the case of the Grover quantum computation [12], this means a node in possession of an actual physical database. Thus we explore for implementation, a three-party distributed computation setting: the client (Alice), who wants to know the answer to a database-lookup problem; the oracle (Oscar), who is in possession of this database and is willing to reveal information about it to a server, but in a blinded fashion that will be intelligible only to Alice; and finally the server itself (Bob), in possession of a powerful quantum computer, with the capacity to receive remote qubits from Alice and Oscar, to perform entangling operations, and to broadcast the results of quantum measurements, under instructions from Alice and Oscar.

Of course, many experiments have achieved some implementation of the two-qubit Grover algorithm. For instance, Refs. [13–15] used the nuclear magnetic resonance (NMR) technique, Refs. [16,17] used trapped ions, Ref. [18] used superconducting qubits, and Ref. [19] used Abelian anyons (in a simulation). Moreover, Refs. [20,21] demonstrated the algorithm with the one-way quantum computer [8,9], and within the blind computation scheme of Ref. [7] was demonstrated in Ref. [22] with just four photonic qubits. But we believe that current developments in the quantum technology of distributed processing using remote NV centers [4,23] make our three-party version of blind client-server quantum computation feasible for a full implementation study.

We can indicate precisely how this implementation can be achieved for the standard two-qubit Grover problem (and will do so in the final section), but we will primarily use the present study to analyze the implementation of scaled-up oracle

*gustiani@physik.rwth-aachen.de

†d.divincenzo@fz-juelich.de

problems. Thus, we will examine in detail the possible realizations of three-qubit Grover. It is already known that going from two to three qubits adds challenges for the implementation: two calls to the oracle are needed rather than one. In addition, the original Grover procedure does not give an error-free identification of the database state, except in the single case of the two-qubit case [24]. This problem was solved by subsequent modifications of Grover's procedure [25–28], and we take account in the present work of these modifications needed to make the database search an "exact" algorithm.

Given the various inconvenient features of three-qubit Grover—two oracle calls, lack of exactness, necessity for two-qubit gates at all stages of the algorithm—it is not surprising that there has been only a limited set of attempts to implement in the laboratory, and never in a distributed or blind setting. Reference [29] illustrated implementation with cavity quantum electrodynamics, an experiment using NMR was performed in Ref. [30], and Ref. [31] demonstrated using trapped atomic ions for different number of queries. However, to maintain the certainty in going from two-bit to three-qubit Grover, more complex gates are required. Only one experiment so far demonstrated the three-qubit exact Grover, which used a magnetic resonance system [32].

But as we show below, "three-qubit Grover" in fact encompasses a very large set of potential algorithms, and we explore these possibilities systematically here, with the objective of identifying the easiest implementations in the NV-center setting. The multiplicities of these Grover algorithms come in several forms. First, the number of database entries can be as many as $N = 2^3 = 8$, but it can be fewer. Each of the new cases $N = 5$, 6, 7, and 8 is a separate problem, and we consider all of these here. While for $N = 8$ all of the three-qubit states are in use, for $N < 8$ only a subset are used; the exact choice of this subset is another variable that we have studied one by one.

There is a final variation of the algorithm that, to our knowledge, has not been exploited before. It is not necessary that the number of distinct entries in the database of Oscar be equal to the number of entries used in the quantum register. For example, suppose that Oscar has five database entries, A, B, C, D, and E. He and Alice may agree on an encoding in which A can correspond to marking either the three-qubit memory location 000 or 001, while the other four have a unique location, say, B $\rightarrow$ 010, C $\rightarrow$ 011, D $\rightarrow$ 100, and E $\rightarrow$ 101. Then, if Alice's final measurement reveals either 000 or 001, she infers that datum A is stored in Oscar's database. The algorithm will also be successful even if Alice cannot reliably distinguish between the 000 and 001 outcomes, so long as they are reliably distinguished from the others. For this reason, we refer to this approach below as the "positive operator-valued measure (POVM) strategy."

We have also exhaustively optimized over possible POVM strategies. We find that the most economical three-qubit Grover algorithm to implement is in fact exactly the one that we have just given as an example! It is perhaps surprising that using $N = 6$ with only five data is preferable to simply using $N = 5$, but we find that the POVM freedom allows for reduction of the gate complexity of the implementation, and thus in the cluster state implementation.

An unfortunate message is that even this most economical case among all the three-qubit Grover algorithms is still much more resource intensive than the two-qubit Grover algorithm. This increase is modest in the number of physical qubits used (four vs three), but very large in the number of gate operations and repeated reuse of physical qubits (approximately $10\times$ more), and correspondingly large in its coherence demands. Thus, it appears that within the Grover family of algorithms, a large jump in the implementation is unavoidable. To make these jumps smaller, it will be necessary to look at other families of oracle algorithms.

II. THEORETICAL BACKGROUND

A. One-way quantum computer

In this study, we consider *one-way quantum computer* (1WQC) [8] as the framework underlying quantum computations. By contrast to conventional quantum computation, *viz.,* the gate model, a 1WQC computation implements unitary maps via a series of projective measurements on a stabilizer state, i.e., *cluster state* [8]. The measurements are done with respect to some ordering and are performed adaptively depending on the previous measurement outcomes. Therefore, in the 1WQC scheme, a *cluster state* defines the quantum computer, and consecutive measurements define quantum operations. Note that we use the convention of Ref. [7] to represent 1WQC computations.

A 1WQC computation can be represented as a set $\{(\mathcal{G}, I, O), \vec{\phi}, \rho\}$, where $\mathcal{G} = (V, E)$ is a connected graph for vertices V and edges E, $I \subset V$ is a set of input nodes, $O \subset V$ is a set of output nodes, ρ is a quantum input assigned to nodes I, and $\vec{\phi}$ is a set of angles parametrizing projective measurements on the nodes, where $\phi_i \in [0, 2\pi)$ for a node i. The cluster state is formed by assigning states to all noninput nodes followed by entangling operations (CPHASE gates) with respect to the edges E. Thus, the cluster state is $\prod_{\{i,j\}\in E} \mathrm{CPHASE}_{i,j} \bigotimes_{i\in I^c} |+\rangle_i$. The measurement projectors have the form $\{|+_\phi\rangle\langle+_\phi|, |-_\phi\rangle\langle-_\phi|\}$, where $|\pm_\phi\rangle := \frac{|0\rangle \pm e^{i\phi}|1\rangle}{\sqrt{2}}$; these are measurements with bases lying on the xy-plane of the Bloch sphere. Note that here we care only about classical inputs and outputs; thus, we measure all nodes in V. A classical input $c_n \ldots c_1 c_0$ is initialized as $\bigotimes_{i\in I} |+_{\pi c_i}\rangle$, where $c_i \in \{0, 1\}$. Moreover, in this study, we consider only input zeros $0 \ldots 0$; thus, we represent a computation as $\{(\mathcal{G}, I, O), \vec{\phi}\}$ instead.

In addition to information $\{(\mathcal{G}, I, O), \vec{\phi}\}$ to represent a computation, we also need a correction scheme. Since quantum measurements unavoidably introduce indeterminacy, adaptive measurements are performed to obtain deterministic quantum operations. A measurement angle ϕ_j can be X- or Z-dependent on previous outcome i, which means correcting ϕ_j to $(-1)^{s_i}\phi_j$ or $\phi_j + s_i\pi$, respectively. Here $s_i \in \{0, 1\}$ is the outcome of measurement i. This correcting scheme is nicely captured by the notion of *flow* [33], that is, a map $f : I^c \mapsto O^c$ following certain criteria (A^c means the complement of set A). Thus, measuring j, $f(j)$ determines X correction, and neighbors of $f(j)$, denoted as $N_{\mathcal{G}}(f(j))$, determine Z corrections. Flow of a graph induces a partial ordering $\succ$ over nodes V, such that the measurement order is consistent with $\succ$.

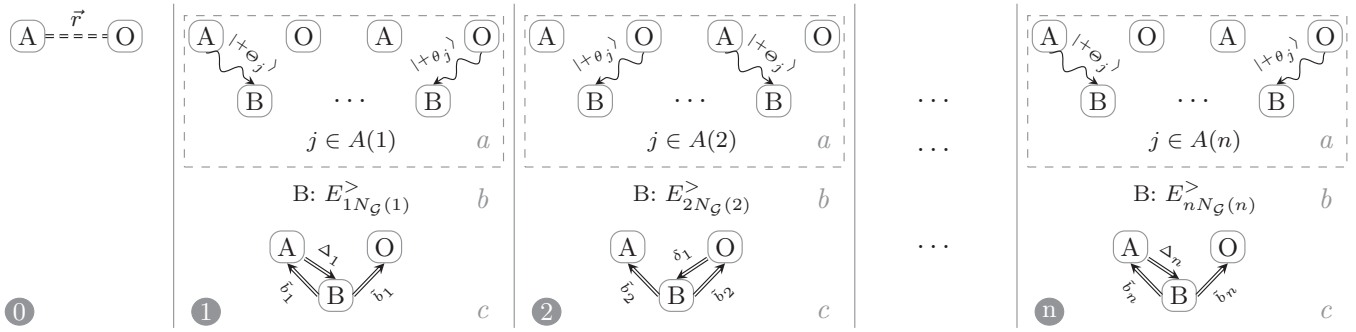


FIG. 1. Communications within the BOQC protocol (Protocol 1 in the Appendix). The protocol takes classical inputs, has classical outputs, and is adapted to Bob's qubits being memory-like (posses permanence and can be rapidly reinitialized). Initials A, B, and O denote Alice, Bob, and Oscar, respectively. The protocol begins with step 0, where Alice and Oscar share a random bit string $\vec{r}$ via a secure key channel (dashed double line). Given that there are n nodes, the computation has n rounds: 1–n. Each round comprises the following: (a) Alice and Oscar transmit the fresh qubits $A(j)$ [Eq. (1)], (b) Bob entangles the corresponding qubits by applying operator $E^{>}_{jN_{\mathcal{G}}(j)}$ [Eq. (2)], and (c) Bob measures qubit j according to the instructed angle (Δ_j or δ_j) and shares his outcome $\tilde{b}_j$ via insecure classical channels (double solid lines).

B. Blind oracular quantum computation

We employ a client-server setting scheme *blind oracular quantum computation* (BOQC) [11] to realize secure oracular quantum computations. BOQC is proven to be composable blind [11], which means the blindness is also maintained as a part of a more extensive cryptographic system. BOQC provides a means to solve the following situation.

Alice is a client who wants to run an oracular quantum algorithm; she has no quantum computer nor the capacity to evaluate her oracle function. Oscar is another client who owns oracles and is willing to cooperate with Alice to run her oracular algorithm. Bob is a server who owns a powerful quantum computer on which Alice and Oscar can run the algorithm. However, Bob is curious, and he is to be prevented ("blinded") from acquiring knowledge of the algorithm or its output. For example, as previously illustrated, in a situation when Alice wants to run a Grover algorithm, and Oscar is in the possession of the database and helps Alice to discover the marked datum in the database by implementing the Grover oracle (or its Høyer variant), without leaking this information to Bob or to any other parties.

There are several variants of the BOQC protocols provided in [11]; we use the one where input and output are classical in which Bob's qubits are memory-like: they possess permanence and can be rapidly reinitialized. For instance, NV-center qubits are memory-like. Note that BOQC protocols that use memory-like qubits are called *BOQC-optimized* (BOQCo) in Ref. [11]. To avoid unnecessary introduction of names, we address BOQCo as BOQC here. For the BOQC variant with classical input-output, the communication resource needed comprises a secure-key channel between Alice and Oscar, one-way quantum channels between Alice or (and) Oscar and Bob, and insecure classical channels between Alice or (and) Oscar and Bob; these resources are illustrated in Fig. 1.

Before running the BOQC protocol, Alice and Oscar perform the following steps, the so-called *preprotocol* [11]. First, Alice and Oscar determine an integer b to construct a set covering all measurement angles: $\Omega = \{\frac{\pi k}{2^{b-1}}\}_{0\leqslant k<2^b}$. Second, given Alice's graph $\mathcal{A}$ and that she needs m oracle queries, she marks the oracles as black boxes. Thus, Oscar's graph, $\mathcal{O}$, is a graph with m components. Oscar sends Alice his graph together with the flow $\{\mathcal{O}, f_{\mathcal{O}}\}$. Alice obtains the whole graph $\mathcal{G} = \mathcal{A} \cup_C \mathcal{O}$ for a connection C, and she computes the total flow $(\succ, f)$. Finally, Alice informs Bob $\{\mathcal{G}, V_{\mathcal{A}}, V_{\mathcal{O}}, \succ, >, b\} =: \ell$, where $V_{\mathcal{A}}$ denotes Alice's nodes, $V_{\mathcal{O}}$ denotes Oscar's node, and $>$ is a total measurement ordering that is consistent with $\succ$. Now every party has the necessary information to run the BOQC scheme.

It turns out that ℓ is the only information that leaks to Bob from the scheme no matter how malicious Bob is. However, knowledge of ℓ restricts Oscar's graph to be identical for all Alice's query. For instance, if Oscar's oracle provides a database with items $\{a, b, c\}$, Oscar's graph $\mathcal{O}$ must be identical for all items a, b, and c. Such a graph is called a *BOQC-compatible graph* in Ref. [11].

Let Alice's computation be $\{\mathcal{A}, \vec{\Phi}\}$ and Oscar's computation be $\{\mathcal{O}, \vec{\phi}\}$; they keep measurement angles (Φ, ϕ) to themselves. Given the total graph $\mathcal{G} = (V, E)$ with a flow $(\succ, f)$, total ordering $>$, input nodes I, and output nodes O, running the computation on the BOQC scheme is done as the following.

The scheme is initiated with Alice and Oscar sharing a random bit string $\vec{r}$, where $r_i \in \{0, 1\}$. All parties run the computation by parts that consist of m rounds, where $m = |V|$. Each round of computation may comprise qubit transmissions, qubit entanglements, and a measurement. Qubit transmissions are done by Alice or Oscar, and Bob performs the rest. Each round i runs as follows.

First, Bob needs to receive fresh qubits that correspond to nodes

$$A(i) := N_{\mathcal{G}}(i) \setminus (\cup_{j<i} N_{\mathcal{G}[j]}) \tag{1}$$

from Alice or (and) Oscar, where $N_{\mathcal{G}}[i]$ indicates *closed neighborhood* of i, namely, nodes adjacent to i including i itself. Let us consider a node $k \in A(i)$. If $k \in V_{\mathcal{A}}$, Alice generates $\theta_k \in \Omega$ at random and sends Bob $\frac{1}{\sqrt{2}}(|0\rangle + e^{i\theta_k}|1\rangle)$; if $k \in V_{\mathcal{O}}$, Oscar generates $\Theta_k \in \Omega$ at random and sends Bob $\frac{1}{\sqrt{2}}(|0\rangle + e^{i\Theta_k}|1\rangle)$. Second, Bob entangles the unmeasured

neighborhood by applying entangling operator

$$E^{>}_{iN_{\mathcal{G}}(i)} \coloneqq \prod_{k\in N_{\mathcal{G}}(i),k>i} E_{ik}, \tag{2}$$

where $N_{\mathcal{G}}(i)$ denotes neighborhood and E_{ik} denotes the CPHASE gate applied to qubits that correspond to nodes i and k. For $i \in V_{\mathcal{A}}$, Alice tells Bob the measurement angle $\Delta_i \coloneqq \Phi'_i + \Theta_i + \pi r_i$; otherwise ($i \in V_{\mathcal{O}}$) Oscar tells Bob $\delta_i \coloneqq \phi'_i + \theta_i + \pi r_i$, where Φ'_i and ϕ'_i are the corrected measurement angles according to the 1WQC scheme. Bob measures qubit i accordingly, then informs his clients upon the measurement outcome $\tilde{s}_i$, where $\tilde{s}_i \in \{0, 1\}$. Finally, Alice and Oscar reveal the correct measurement outcome $s_i = \tilde{s}_i \oplus r_i$. We provide an explicit form of this protocol in the Appendix in Protocol 1.

Note that the hiding protocol of BOQC is adapted from the Universal Blind Quantum Computation (UBQC) scheme introduced in [7]. However, the UBQC has one client instead of two. The BOQC promises the same security as UBQC with minimal communication between clients, which is almost none. Communications among players in the BOQC protocol are illustrated in Fig. 1. The figure shows that Alice and Oscar communicate only in the beginning, namely, via a secure key channel.

C. The exact Grover-Høyer search algorithm

The optimality of the Grover algorithm is well known [34]; high success probability is achieved with the fewest iterations. As the number of items in the database N increases, the success probability approaches one, whereas for small N the error is appreciable. For instance, success probabilities (p_N) running three-qubit Grover are $p_5 = 0.968$ with one iteration, $p_6 = 0.907$ with one iteration, $p_7 = 0.871$ with two iterations, and $p_8 = 0.945$ with two iterations. Because of this problem, many workers devised modifications or generalizations of the Grover algorithm to achieve probability one. For instance, Chi and Kim [25] proposed a single query search for cases when marking one-quarter of the database, and Høyer [26] introduced arbitrary phase rotation in quantum amplitude amplification; while Høyer's phase matching condition is approximate without manipulating the initial condition, Long *et al.* [35] preceded an exact phase matching condition, which then generalized in Ref. [36] where Høyer's condition can be acquired. Long also contributed an exact quantum search algorithm in Ref. [27] for an arbitrary size of databases and multiple marked items. Much later, Liu [28] also proposed the generalization for an arbitrary size and combination of databases. This section provides details of the so-called *Grover-Høyer algorithm*, which combines previous Grover and Høyer procedures to achieve probability one, and later we develop an algorithm based on that, which also features oracle separation, blindness, and measurement freedom.

Suppose n qubits are used to represent all indices $x = \{0, \ldots, 2^{n-1}\}$. One may arbitrarily choose N elements of x that represent indices of a database w, thus $w \subset x$, where $|w| = N$, and we will consider the case $2^{n-1} < N \leqslant 2^n$. Without loss of generality, we start from a product of zero states $|0\rangle^{\otimes n}$. We consider an operator A that maps a product state into an equal superposition of N states, thus $A|0\rangle^{\otimes n} = (1/\sqrt{N})/\sum_{j\in w}|j\rangle =: |\Psi_{in}\rangle$. Suppose we have marked items in the database $\tau \subset w$—we are interested in a special case where $|\tau| = 1$, thus $\tau \in w$. Given an oracle that evaluates a function $f(j)$ that indicates if j indexes a marked item of database, f induces a partition in the Hilbert space into "solutions" (τ) and "nonsolutions" ($w \setminus \tau$) subspaces. Rewrite the state $|\Psi_{\text{in}}\rangle = \sqrt{a}|\tilde{\Psi}_1\rangle + \sqrt{1-a}|\tilde{\Psi}_0\rangle$, where $a = 1/N$ and $|\tilde{\Psi}_1\rangle$ and $|\tilde{\Psi}_0\rangle$ are the normalized states corresponding to $|\Psi_1\rangle \coloneqq \sum_{j\in\tau}|j\rangle$ and $|\Psi_0\rangle \coloneqq \sum_{j\in w\setminus\tau}|j\rangle$. Henceforth, we will work in the Hilbert space defined as the subspace spanned by basis $\{|\tilde{\Psi}_0\rangle, |\tilde{\Psi}_1\rangle\}$.

Algorithm 1. Grover-Høyer algorithm

Require: w, τ
(1) Classical processing
$N \leftarrow$ size of w
Ensure: $2^{n-1} < N \leq 2^n$ and $\tau \in w$
$\theta_0 \leftarrow \arcsin(1/\sqrt{N})$
$m \leftarrow \lfloor(\frac{\pi}{2} - \theta_0)/2\theta_0\rfloor$ number of Grover runs
$\theta \leftarrow \frac{\pi}{2} - 2m\theta_0 - \theta_0$ the remaining rotation
$\psi, \varphi, u \leftarrow$ Equation (6).
$A \leftarrow |\Psi_{in}\rangle\,(\langle 0|)^{\otimes n}$
$\{D(\pi), D(\psi)\} \leftarrow$ Equation (4)
$\{O(\pi), O(\varphi + u)\} \leftarrow$ Equation (3)
Obtain a set of necessary operators $\mathcal{B}$ (Equation (5)), which are expressed within operations that can be done with the corresponding quantum computer.
(2) Quantum processing
$|\Psi\rangle \leftarrow \mathcal{A}\,|0\rangle^{\otimes n}$
for $j = 1$ to m **do**
  $|\Psi\rangle \leftarrow \mathcal{D}(\pi)\mathcal{O}(\pi)\,|\Psi\rangle$,
end for
$|\Psi\rangle \leftarrow \mathcal{D}(\psi)\mathcal{O}(\varphi + u)\,|\Psi\rangle$
Measure $|\Psi\rangle$
Exit

Using previously described variables, running Algorithm I within database w will reveal the marked item τ with probability one. The main idea of the algorithm is to combine the Grover algorithm with Høyer's arbitrary phase rotation (also known as Høyer amplitude amplification), which performs the necessary rotation to bring the state vector exactly into the solution space. The modified iteration introduces new operators $\{O(\varphi + u), D(\psi)\}$, where

$$O(\varphi) = -I + (1 - e^{i\varphi})|\tau\rangle\langle\tau|, \tag{3}$$

$$D(\psi) = -I + (1 - e^{i\psi})|\Psi_{\text{in}}\rangle\langle\Psi_{\text{in}}|. \tag{4}$$

The algorithm comprises two stages: *classical processing*, where a compatible set of operations for every required unitary is obtained:

$$\{\mathcal{A}, \mathcal{O}(\pi), \mathcal{D}(\pi), \mathcal{O}(\varphi + u), \mathcal{D}(\psi)\} =: \mathcal{B}, \tag{5}$$

and *quantum processing*, where the quantum computation is performed on the quantum computer; every operator in $\mathcal{B}$ respectively corresponds to unitary matrices in $\{A, O(\pi), D(\pi), O(\varphi + u), D(\psi)\}$. When we say that we have a compatible set of operations $\mathcal{A}$ corresponding to the unitary operator A (and similarly for all elements of $\mathcal{B}$), we mean that we specify an explicit implementation of A as a sequence

of operations $\mathcal{A}$ that can be performed for some model of quantum computation, e.g., in the form of quantum gates or operations on a cluster state. For instance, our result in Fig. 3 works on a quantum computer which performs CNOT and arbitrary one-qubit gates.

The Høyer amplitude amplification is described by an operator $Q(\varphi, \psi) = D(\psi)O(\varphi)$, which rotates a state closer to the solution space by as much as θ, where $|\sin(\theta)| \leqslant \sin(2\theta_0)$, $\theta_0 = \arcsin(1/\sqrt{N})$. Høyer found φ and ψ such that $Q(\varphi, \psi)$ performs the desired rotation:

$$\begin{aligned}\psi &= \arccos\left(1 - \frac{\sin^2(\theta)}{2a(1-a)}\right),\\ \varphi &= 2\arctan[\tan(\psi/2)(1-2a)],\\ u &= \arg((1-e^{i\psi})\sqrt{a(1-a)})\\ &\quad - \arg(-a(1-e^{i\psi}) - e^{i\psi}).\end{aligned} \tag{6}$$

Using those angles, $Q(\varphi, \psi)$ rotates the state by angle θ up to some phases $\pm u$:

$$Q(\varphi, \psi) = \begin{pmatrix}1 & 0\\ 0 & e^{iu}\end{pmatrix}\begin{pmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{pmatrix}\begin{pmatrix}1 & 0\\ 0 & e^{-iu}\end{pmatrix}. \tag{7}$$

The unwanted phases $\pm u$ can be canceled by performing the sequence $P(-u)Q(\varphi, \psi)P(u)$, where $P(\alpha) = -I + (1 - e^{i\alpha})|\tau\rangle\langle\tau|$. Since the form of operator P is identical to that of O, $O(\varphi)P(u) = O(\varphi + u)$ (see step 14 of Algorithm 1). When Høyer amplitude amplification is applied in the last iteration of the Grover algorithm, the state is entirely aligned to the solution space after the application of $P(-u)Q(\varphi, \psi)$. Thus, applying $P(u)$ afterward will change only the global phase of the state. This is the reason for omitting the last phase correction in Algorithm 1. Note that Algorithm 1 is an explicit implementation of the exact search algorithm mentioned by Høyer in Ref. [26].

III. NUMERICAL SEARCH AND OPTIMIZATION

A. Exhaustive search for the most economical exact Grover algorithm

The challenge in realizing the Grover-Høyer algorithm—apart from running the quantum processing with arbitrarily small error—is the optimization of the circuit preparation indicated on line 9 of Algorithm 1, where the desired unitary map must be written out as a set of quantum gates that can be run in the quantum computer. We develop an approach based on DiVincenzo and Smolin [37] (DS94) to overcome this challenge; such a challenge will appear again later when we need to obtain a graph state. This section mainly reviews DS94.

DS94 is a systematic, exhaustive approach: given the desired unitary map M, where $M \in$ SU(8), a set of two-qubit gates networks are optimized over, where every two-qubit gate is in SU(4). We refer to "topology" of a two-qubit gate network as a configuration of those two-qubit gates. As we are concerned here with a three-qubit operations, as was also the case in the study of DS94, the notations of DS94 are used: qubits are indicated with numbers 1, 2, and 3; a two-qubit gate is indicated with the number of the untouched qubit. A topology is denoted by numbers within parentheses, where each number represents the corresponding two-qubit gate. So, for example, topology (321) indicates two-qubit gates applied on qubits: {1, 2}, {1, 3}, and {2, 3}; note that the order of gates here is relevant, since these gates do not commute.

Algorithm 2. Circuit search

Require: w, τ
1: $l \leftarrow 0$
2: $\mathcal{N} \leftarrow$ all unique topologies of 2-qubit gates network with size l
3: **for** G in $\mathcal{N}$ **do**
4: Optimize G
5: **if** the optimization succeeds **then return** G, the circuit is found and **Exit**
6: **end if**
7: **end for**
8: $l \leftarrow l + 1$ and go to line 2

To obtain an exhaustive set of topologies, all possible topologies of two-qubit gate networks are enumerated, then the equivalent ones are eliminated. Two different topologies can be equivalent for the following reasons [37]: *time-reversal,* which means placing the gates in time-reversed order, e.g., (12123) = (32121); *bit-relabeling,* e.g., relabeling qubit 1 and 2, thus (12123) = (21213); and *conjugation by swapping,* which means swapping of the states of any pair of bits, e.g., (12123) = (13123) = (12323) = (12313). For the systems with an unused subspace in the Hilbert space—thus for $N < 8$, where N is the dimension of the Hilbert space—the reordering must preserve the state space. For instance, database $w = \{0, 1, 2, 4, 7\}$ is conserved with permutation of every element in S_3; this is easiest seen by writing this set w in three-bit notation, $w = \{000, 001, 010, 100, 111\}$. On the other hand, database $w = \{0, 1, 2, 3, 4\}$ is conserved only with one permutation of S_3: $\left(\begin{smallmatrix}1 & 2 & 3\\ 1 & 3 & 2\end{smallmatrix}\right)$.

The nonlinear minimization Broyden-Fletcher-Goldfarb-Shanno (BFGS) [38] is used for the optimization in DS94 with the objective function defined as $f = \sum_i \sum_j |M_{ij} - S_{ij}|^2$, where M is the desired SU(8) unitary, and S is the matrix resulting from composing the two-qubit gate network. The minimization is over the parameters of the individual SU(4) matrices describing the two-qubit gates. It is successful if $f = 0$ to a reasonable accuracy; thus a two-qubit gate network that implements M is found.

B. Circuits for Grover-Høyer algorithm

In this section, we present a strategy to obtain quantum circuits that run the Grover-Høyer algorithm. We will specifically explore cases where the database is encoded within three qubits, where $N = 5, 6, 7,$ and 8. The strategy essentially is seeking every circuit in $\mathcal{B}$ [Eq. (5)] using DS94 optimization. The main challenges are the abundance of database choices and two-qubit gate networks to be tried; note that $\binom{8}{N}$ database choices are possible for each N. A strategy to group those choices into a small number of equivalent sets will also be presented here.

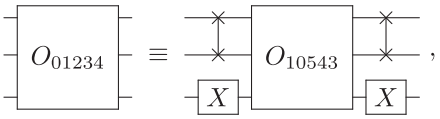
We seek quantum circuits using Algorithm 2—for $N \in \{5, 6, 7, 8\}$, for all unique database combinations w, and all marked items $\tau \in w$—by finding all operations in $\mathcal{B}_{w,\tau}$ [see Eq. (5)], that is, the required operators to run the Grover-

Høyer algorithm for a database w and a marked item τ. Note that this search of circuits is done separately—one may do it for the whole Grover-Høyer algorithm and obtain smaller circuits—in order to obtain a BOQC-compatible circuit.

We will see that many database choices are equivalent by considering the role of the Grover oracle. For convenience, rewrite a database set $w = \{d_1, d_2, \ldots, d_N\} \equiv d_1 d_2 \ldots d_N$, where $d_j \in \{0, 1, 2, 3, 4, 5, 6, 7\}$. One is given three bits $|ijk\rangle$ to encode w, where $i, j, k \in \{0, 1\}$, and a set of oracle operators where each of them "marks" one element by phase $e^{i\varphi}$. Two sets of database w_1, w_2, where $|w_1| = |w_2|$, are equivalent if a set of oracles that can mark for all $\tau_1 \in w_1$ can also mark for all $\tau_2 \in w_2$ up to some global phases.

By this means, while considering their bit representations, w_1 is equivalent to w_2 if they are identical up to permutation and bit complementation. For instance, consider two equivalent databases with their bit representations (in little-endian format): 01234 = {000, 001, 010, 011, 100}, 10543 = {001, 000, 101, 100, 011}. One can be obtained from another by complementing the third bit and permuting the first and the second bits.

In the gate model, it means their oracles are equivalent up to some operations:



(8)

where O_{01234} and O_{10543} represent the oracle operators for databases 01234 and 10543 respectively. With these equivalences, all databases are covered by the set $\mathcal{D} :=$ {01234, 01247, 01256, 012345, 012347, 012567, 0123456, 01234567}. Set $\mathcal{D}$ is obtained by picking the smallest partition of all possible five-database sets (see Table III in the Appendix for the complete partitions of all possible 3-qubit databases). Note that this strategy works for an arbitrary number of bits, not only for three.

As one may freely define a set of quantum gates that compose gate networks [for instance, DS94 considered the set of all U(4) matrices], we compose the gate networks into the operations $\{\text{CNOT}, U(\alpha, \beta, \gamma)\}_{\alpha,\beta,\gamma\in[0,2\pi)}$, where $U(\alpha, \beta, \gamma)$ is a unitary matrix drawn from a family of gates in SU(2) that has the form

$$U(\alpha, \beta, \gamma) = \begin{pmatrix} e^{i\beta}\cos(\alpha) & e^{i\gamma}\sin(\alpha) \\ -e^{-i\gamma}\sin(\alpha) & e^{-i\beta}\cos(\alpha) \end{pmatrix}, \tag{9}$$

where $\alpha, \beta, \gamma \in [0, 2\pi)$ are free parameters; these will be the optimization parameters below. We define l to be the number of CNOTs in our three-qubit network. For $l = 0$, the network is simply three one-qubit gates; for every additional CNOT gate, two one-qubit gates are added after. Thus, $6l + 9$ free parameters will be available for the optimization for a network with size l. All networks for $l = 0$ and $l = 1$ are shown in Fig. 2.

Again, not all networks are distinct; we obtain a minimal set of representative networks by enumerating all possible network topologies, followed by two eliminations: we eliminate ones that have more than three consecutive CNOT gates, and we eliminate the ones that are topologically equivalent [37] (also discussed in Sec. III A). The first elimination is based

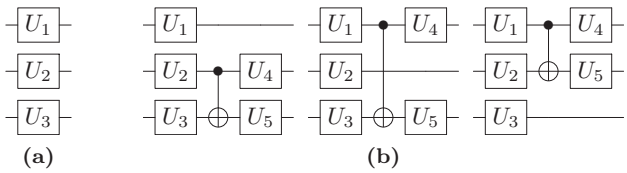


FIG. 2. Two-qubit gate networks for $l = 0$ and $l = 1$. (a) Network size $l = 0$ has topology (0). (b) Network size $l = 1$ has topologies (1), (2), and (3), shown from the left to the right, respectively. Operators $U_j \equiv U_j(\alpha_j, \beta_j, \gamma_j)$ are one-qubit gates as in Eq. (9).

on the fact that an arbitrary SU(8) can be constructed using three CNOT gates and eight one-qubit gates [39]. Thus, for example, topology (133331) is eliminated since (133331) = (13331).

We use DS94 optimization within Algorithm 2 to find the gate networks. A BFGS solver of the Python SciPy library [40] is employed in our program. To speed up optimizations, we define more relaxed objective functions than DS94:

$$f_b = \sum_{i,j\in w, M_{ij}\neq 0} |M_{ij} - S_{ij}|^2,$$
$$f_p = \sum_{i\in w, M_{i0}\neq 0} |M_{i0} - S_{i0}|^2, \tag{10}$$

where M is the desired unitary matrix and S is the resulting matrix from the tested network (G); f_p is used if $M = A$; that is, the preparation block in $\mathcal{B}_{w,\tau}$, and f_b is used for other blocks in $\mathcal{B}_{w,\tau} \setminus \{A\}$. While the Ms are assumed to be unitary matrices here, it is sufficient to consider only the nonzero elements within the subspace that is induced by w. Note that f_p is appropriate for $M = A$ because we start from the all-zero state $|0\rangle^{\otimes n}$.

Success in optimization is defined as $f_p \leqslant \varepsilon$ or $f_b \leqslant \varepsilon$, where ε is a chosen error bound. We define ε such that the success probability is approximately one: given δ, there exists an ε such that $p_s \geqslant 1 - \delta$, where p_s is the success probability of running Algorithm 1 while replacing block M with the tested network G. For the nonoracle cases, $M \neq O$, we take the worst p_s among all obtained p_s from different marked items.

Table I shows the size of the network for every operator in $\mathcal{B}_{w,\tau}$, for all unique database sets $w \in \mathcal{D}$, and for all valid marked items where $\delta = 10^{-4}$. We obtain $\varepsilon_p \leqslant 4.8\times10^{-11}$ for preparation blocks and $\varepsilon_b \leqslant 2.5\times10^{-8}$ for other blocks. The complete tables that show values of success probabilities p_s are shown in the Appendix, Table IV. While this does not complete our analysis of the three-qubit Grover algorithms, these preliminary calculations indicate that the most efficient network will be achieved for $N = 6$ and $w = 012345$ (and not the smaller $N = 5$).

At this point, we complete the classical processing stage of Table I. Since the blocks are prepared independently, this result can be adapted to develop the full BOQC scheme. However, for $N < 8$, the straightforward implementation of the oracles would require different network sizes for different marked items τ. This would allow Bob to learn about Alice's request to Oscar. In the next section, we complete our exact quantum search algorithm, taking care that networks of

TABLE I. The number of CNOT gates for all distinct combinations of (w, τ, M) for all $w \in \mathcal{D}$, $\tau \in w$, $M \in \{A, O_w(\pi), D(\pi), O_w(\phi + u), D(\psi)\}$, and $N \in \{5, 6, 7, 8\}$. The angles φ, ψ, and u refer to the Høyer exact-Grover technique; see Algorithm 1. The boldface indicates the least number of CNOT gates among the combinations, where the oracles with the largest number of CNOT gates are taken into account.

w	A				$O(\pi)$			$D(\pi)$				$O(\varphi+u)$				$D(\psi)$
							$[N=5, \varphi+u=0.1707, \psi=0.4510]$									
01234		0	1	2	3	4		0	1	2	3	4				
CNOT	2	1	1	1	1	0	7	2	2	2	2	0		8		
01247		0	1	2	4	7		0	1	2	4	7				
CNOT	2	0	1	1	1	1	7	0	2	2	2	2		8		
01256		0	1	2	5	6		0	1	2	5	6				
CNOT	3	0	1	1	1	1	8	0	2	2	2	2		9		
							$[N=6, \varphi+u=1.861, \psi=0.841]$									
012345		**0**	**1**	**2**	**3**	**4**	**5**		**0**	**1**	**2**	**3**	**4**	**5**		
CNOT	**1**	**2**	**2**	**1**	**1**	**1**	**1**	**4**	**4**	**4**	**2**	**2**	**2**	**2**	**6**	
012347		0	1	2	3	4	7		0	1	2	3	4	7		
CNOT	2	2	1	1	2	1	1	6	4	2	2	4	2	2	7	
012567		0	1	2	5	6	7		0	1	2	5	6	7		
CNOT	3	1	1	1	2	1	1	8	2	2	2	2	2	2	8	
							$[N=7, \varphi+u=2.0277, \psi=1.2056]$									
0123456		0	1	2	3	4	5	6		0	1	2	3	4	5	6
CNOT	3	3	2	2	1	2	1	1	8	4	4	6	2	4	2	2	9
							$[N=8, \varphi+u=2.2143, \psi=1.5708]$									
01234567		0	1	2	3	4	5	6	7		0	1	2	3	4	5	6	7	
CNOT	0	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6

identical structure are created for each value of τ, ensuring the blindness of the protocol.

C. The exact quantum search algorithm with blind oracles

Here we introduce an algorithm called the *blind exact quantum search algorithm* (BEQS), given in Algorithm 3, which is an improvement of the Grover-Høyer algorithm: it is compatible with the BOQC scheme, and it involves a general scheme for storing information in the database. Achieving the first means obtaining identical oracles for all marked items, whose measurement angles are adjusted accordingly. The latter means permitting several marked items τ to stand for a single database entry; we do this by making the final measurement of the Grover algorithm an incomplete or POVM measurement. For instance, consider two-bit Grover algorithm with database $w = 0123$, $N = 4$, and database entries {A,B} ($\tilde{N} = 2$). Note that here we distinguish between the database size (N) and the number of database entries ($\tilde{N}$). Alice and Oscar agree ahead of time that either outcome 0 or 1 corresponds to entry A, and outcome 2 or 3 corresponds to entry B; this is attainable by defining measurement operators $\{|0\rangle\langle 0| + |1\rangle\langle 1|, |2\rangle\langle 2| + |3\rangle\langle 3|\}$. We will refer to such a scheme as a "POVM measurement strategy." Physically, it is possible to still do the full projective measurement, then classically associate the measurement outcomes with database entries.

While it is hard analytically to obtain an identical form—in our case using gate networks—of oracles for all marked items, the BEQS provides a numerical method to obtain all those oracles using a single numerical procedure. The key lies in the objective function ObjPOVM in Subroutine 1, which includes two constraints: (C1) the success probability must be one, and (C2) the resulting operator must preserve the

Algorithm 3. Blind exact quantum search

Require: n, w, τ, M_{POVM}

(1.a) Classical processing done by Alice

1: Prepare $\{\mathcal{A}, \mathcal{D}(\pi), \mathcal{D}(\psi)\}$ (Equation (5)) using Algorithm 2, with objective functions f_a or f_b (Equation (10)).

(1.b) Classical processing done by Oscar

2: Search for the blind oracles $\tilde{\mathcal{O}}$ using Algorithm 2 with the objective function

$$\text{ObjPOVM}(G_j, M_{POVM}, \vec{\varphi}, n, m, w, A, D(\pi), D(\psi)),$$

defined in Subroutine 1, where G_j is the tested network, $\vec{\varphi} \equiv (\vec{\varphi}_1, \ldots, \vec{\varphi}_N)$ is the optimization parameters — with random initialization — for all marked items $\tau \in w$, M_{POVM} is the defined POVM, and the rest $(n, m, w, A, D(\pi), D(\psi))$ are defined as those in Algorithm 1. If the optimization succeeds, set $\tilde{\mathcal{O}} \leftarrow G_j$ and $\vec{\varphi}$ is optimized. All oracles now have the same networks, but different parameters, which are set according to the marked item.

3: Set $\mathcal{O}(\pi) \leftarrow \tilde{\mathcal{O}}$ and $\mathcal{O}(\varphi + u) \leftarrow \tilde{\mathcal{O}}$; the two oracle calls labelled $\mathcal{O}(\pi)$ and $\mathcal{O}(\varphi + u)$ in previous algorithms will be accomplished by the same oracle operation $\tilde{\mathcal{O}}$.

(2) Quantum processing

4: Perform the quantum processing of Algorithm 1 within the respective scheme. If the scheme is BOQC, run it according to the scheme of Figure 1.

5: **Exit**

Subroutine 1. OBJPOVM

Require: $G_j, M_{POVM}, \vec{\varphi}, n, m, w, A, D(\pi), D(\psi)$

1: $ov \leftarrow 0$, the objective value
2: **for** τ in w **do**
3: $S \leftarrow G_j(\vec{\varphi}_\tau)$
4: $|\Psi\rangle \leftarrow A|0\rangle^{\otimes n}$
5: **for** $i = 1$ to m **do**
6: $|\Psi\rangle \leftarrow D(\pi)S|\Psi\rangle$
7: **end for**
8: $|\Psi\rangle \leftarrow D(\psi)S|\Psi\rangle$
9: $ov \leftarrow ov + 1 - |\langle\tau| M_{POVM}(\tau)|\Psi\rangle|$
10: $ov \leftarrow ov + |f_{obd}|$ (Equation (11))
11: **end for**
12: **return** ov

state space. Recall that database choice w induces the state space.

The first constraint, C1, implemented at line 9 of Subroutine 1, imposes a successful computation within the defined POVM measurement for all permitted marked items—notice that the loop goes for all $\tau \in w$. Constraint C2, implemented at line 9 of Subroutine 1, ensures a block diagonal matrix, which is critical when there is a free subspace in the full 2^n-dimensional Hilbert space for an n-qubit system. This constraint is imposed by requiring that the sum of the absolute values of the elements outside diagonal block be zero:

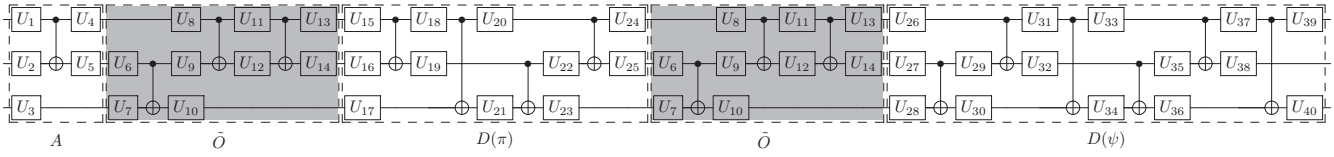
$$f_{obd} = \sum_{i\in x, j\in x\setminus w, i\neq j} |S_{ij}|^2 + \sum_{i\in x\setminus w, j\in w} |S_{ij}|^2, \qquad (11)$$

where w is the database index, x are all possible indices that can be accommodated, and S is the matrix from evaluating a network $G_j(\vec{\varphi}_\tau)$. All constraints are quantified within the objective value ov. It is worth mentioning that the obtained operator $\tilde{O}$ forms a block diagonal matrix that necessarily resembles neither the Grover nor the Høyer oracles.

Unfortunately, OBJPOVM requires more resources than f_p and f_b [Eq. (10)]; therefore for reasons of economy, we set a fixed oracle in every query, which results in fewer optimization parameters. One possible improvement is restricting the legitimate marked items $q \subset w$ to cut the loop at line 2 of Subroutine 1. Returning to the previous example where $w = 0123$, $M_{\text{POVM}} = \{|0\rangle\langle 0| + |1\rangle\langle 1|, |2\rangle\langle 2| + |3\rangle\langle 3|\}$, and the database entries are {A,B}, we simply set $q = \{0, 2\}$. Whereas previously Oscar would randomly mark item 0 or 1 to reveal A and would randomly mark 2 or 3 to reveal B, now Oscar marks only 0 to reveal A and marks 2 to reveal B. This amount of speedup resulting from this strategy depends on how small q compared to w.

We test the BEQS for three-qubit cases, obtaining quantum algorithms within the gate model and the 1WQC model (this takes care of the BOQC model also), where $w = 012345$ and $M_{\text{POVM}} = \{|0\rangle\langle 0| + |1\rangle\langle 1|, |2\rangle\langle 2|, |3\rangle\langle 3|, |4\rangle\langle 4|, |5\rangle\langle 5|\}$, thus $N = 6$ and $\tilde{N} = 5$. We choose this configuration based on its potential to result in the smallest gate network based on the study of Table I. We obtain Fig. 3 for the gate model, that is, a circuit comprising $\{\text{CNOT}, U(\alpha, \beta, \gamma)\}_{\alpha,\beta,\gamma\in[0,2\pi)}$, where $U(\alpha, \beta, \gamma)$ has a form of Eq. (9). Our result in Fig. 3 might appear to be overparameterized for an operation SU(8); however, one must consider the unitary as an individual in $\{\mathcal{A}, \tilde{O}, D(\pi), \tilde{O}, D(\psi)\}$, because of the blindness requirement in preparing a BOQC-compatible graph.

To enable direct conversion of quantum gates into a graph state, we perform another optimization that decomposes the result in Fig. 3 into another set of gates, namely, $\{\text{CPHASE}, H, R_z(\alpha)\}_{\alpha\in[0,2\pi)}$, where $R_z(\alpha) =$



Gate	α	β	γ
U_1	2.1863	3.4700	3.4700
U_2	2.5159	3.3937	2.1618
U_3	5.4978	4.7124	1.5708
U_4	1.5708	1.8160	5.0408
U_5	5.6584	2.8913	5.3000
U_{15}	0.3929	1.9270	1.9237
U_{16}	0.7011	2.8271	1.0836

U_{17}	4.7124	2.7614	1.5708
U_{18}	5.0204	1.3905	5.2465
U_{19}	0.8541	2.7177	2.1142
U_{20}	2.8341	0.3039	3.7996
U_{21}	1.5708	4.2914	1.5708
U_{22}	5.5189	5.6770	4.3625

U_{23}	0.0000	3.1416	4.0531
U_{24}	1.9641	2.6566	0.4818
U_{25}	3.9010	1.9380	5.8626
U_{26}	4.3149	5.8000	2.8293
U_{27}	4.1749	1.0410	4.8345
U_{28}	3.3519	4.7124	0.0000

U_{29}	5.3865	0.4235	2.8814
U_{30}	2.5762	0.2296	4.8464
U_{31}	2.3073	5.0124	4.7439
U_{32}	1.5708	3.3484	3.1416
U_{33}	1.9640	4.5092	3.5709
U_{34}	0.9027	4.5151	4.5872

U_{35}	2.2493	2.3631	0.7364
U_{36}	3.1416	3.1416	5.0640
U_{37}	5.5892	4.5517	5.0515
U_{38}	1.0992	1.0522	2.5764
U_{39}	4.2518	5.6790	6.1262
U_{40}	1.4576	1.4573	4.4534

$\tau = 0$			
U_6	0.0000	5.2267	2.4690
U_7	0.0000	4.7124	2.3946
U_8	0.8805	5.0403	1.9146
U_9	2.9266	0.4472	2.0889
U_{10}	4.7124	4.6003	1.5708
U_{11}	5.2427	0.2804	1.5793
U_{12}	4.9284	3.4651	4.2091
U_{13}	2.5244	4.1733	4.6219
U_{14}	3.8309	4.1770	2.3886

$\tau = 1$			
U_6	0.0000	2.2951	1.0317
U_7	4.7124	4.5527	3.1416
U_8	3.1416	4.4256	3.1266
U_9	1.5234	1.1716	0.8384
U_{10}	3.1416	4.7124	3.7358
U_{11}	0.0000	4.0069	2.5518
U_{12}	2.7625	2.4785	4.1532
U_{13}	0.0000	5.0886	2.9811
U_{14}	5.1326	4.2242	0.7998

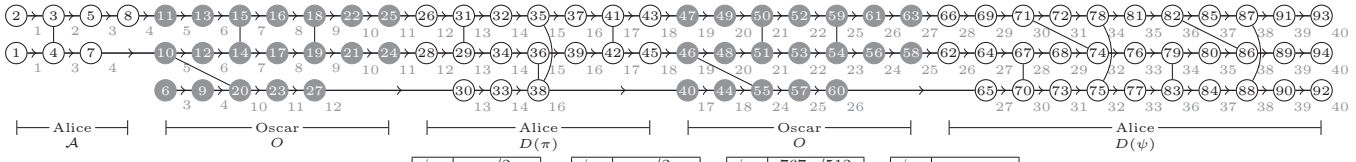
$\tau = 2$			
U_6	1.5708	5.5013	2.2216
U_7	0.1287	3.2740	5.7062
U_8	4.1648	3.5317	5.6437
U_9	2.7877	5.7409	5.7010
U_{10}	6.1545	2.4033	5.1003
U_{11}	1.9797	5.6191	1.5769
U_{12}	4.9201	2.2995	2.3634
U_{13}	5.7356	4.4452	2.8675
U_{14}	1.7632	0.0188	3.8868

$\tau = 3$			
U_6	1.5708	0.7393	1.9242
U_7	5.3911	3.3728	2.8631
U_8	2.4257	0.0141	1.7221
U_9	0.3262	3.4474	4.6704
U_{10}	5.3911	1.5255	4.6197
U_{11}	5.6475	2.6774	1.5045
U_{12}	4.1456	4.2234	4.1602
U_{13}	0.6915	1.0830	2.5152
U_{14}	6.2190	0.9305	3.3521

$\tau = 4$			
U_6	0.0000	1.8189	4.4016
U_7	1.7606	5.7840	5.8093
U_8	2.3289	0.1070	3.0468
U_9	2.9211	1.0980	3.6533
U_{10}	6.0934	5.6116	3.4430
U_{11}	1.9619	1.6645	2.0202
U_{12}	3.6540	3.1061	0.6672
U_{13}	5.5116	5.2363	5.8918
U_{14}	4.7451	4.3359	0.4896

$\tau = 5$			
U_6	0.0000	2.7690	1.7284
U_7	5.2442	3.6695	5.6213
U_8	3.8175	6.0993	4.1414
U_9	0.3776	5.1356	5.8965
U_{10}	3.6734	2.9525	6.2280
U_{11}	2.6959	3.5218	0.4414
U_{12}	1.2797	5.6970	2.5852
U_{13}	5.5732	0.2160	0.9320
U_{14}	1.6530	0.2316	0.0389

FIG. 3. A three-qubit BEQS (see Algorithm 3) within the gate model for $w = 012345$ and $M_{\text{POVM}} = \{|0\rangle\langle 0| + |1\rangle\langle 1|, |2\rangle\langle 2|, |3\rangle\langle 3|, |4\rangle\langle 4|, |5\rangle\langle 5|\}$. Outcomes 0 and 1 refer to the same data, thus $N = 6$ and $\tilde{N} = 5$. The circuit is composed of $\{\text{CNOT}, U(\alpha, \beta, \gamma)\}_{\alpha,\beta,\gamma\in[0,2\pi)}$, where $U(\alpha, \beta, \gamma)$ is a SU(2) matrix and has the form of Eq. (9). The gray blocks indicate blind oracles; their parameters (see tables below) are different for each marked item τ. The circuit has success probabilities $p_s \geqslant 1 - 10^{-4}$ for all marked items $\tau \in w$.



parameter	angles
ϕ_1	$213\pi/512$
ϕ_2	$97\pi/128$
ϕ_3	$471\pi/256$
ϕ_4	$719\pi/512$
ϕ_5	$155\pi/128$
ϕ_7	$915\pi/512$
ϕ_8	$299\pi/512$
ϕ_{26}	$3\pi/2$

ϕ_{28}	$\pi/2$
ϕ_{29}	$\pi/4$
ϕ_{30}	$\pi/2$
ϕ_{31}	$3\pi/4$
ϕ_{32}	$39\pi/128$
ϕ_{33}	$\pi/2$
ϕ_{34}	$39\pi/128$
ϕ_{35}	π
ϕ_{36}	π
ϕ_{37}	$217\pi/128$

ϕ_{38}	$\pi/2$
ϕ_{39}	$89\pi/128$
ϕ_{41}	$7\pi/4$
ϕ_{42}	$7\pi/4$
ϕ_{43}	$3\pi/2$
ϕ_{45}	$3\pi/2$
ϕ_{62}	$129\pi/256$
ϕ_{64}	$\pi/4$
ϕ_{65}	0
ϕ_{66}	$535\pi/512$

ϕ_{67}	$767\pi/512$
ϕ_{68}	$991\pi/512$
ϕ_{69}	$281\pi/512$
ϕ_{70}	$613\pi/512$
ϕ_{71}	$387\pi/512$
ϕ_{72}	$467\pi/512$
ϕ_{73}	π
ϕ_{74}	$255\pi/512$
ϕ_{75}	$125\pi/256$
ϕ_{76}	$1001\pi/512$

ϕ_{77}	π
ϕ_{78}	$\pi/256$
ϕ_{79}	π
ϕ_{80}	$535\pi/512$
ϕ_{81}	$45\pi/512$
ϕ_{82}	$3\pi/2$
ϕ_{83}	$495\pi/512$
ϕ_{84}	π
ϕ_{85}	$479\pi/512$
ϕ_{86}	$497\pi/256$

ϕ_{87}	$641\pi/512$
ϕ_{88}	$313\pi/256$
ϕ_{89}	$643\pi/512$
ϕ_{90}	π
ϕ_{91}	$\pi/2$
ϕ_{92}	$53\pi/128$
ϕ_{93}	$257\pi/256$
ϕ_{94}	$53\pi/128$

$\tau = 0$	
$\phi_{11/47}$	$249\pi/512$
$\phi_{6/40}$	$275\pi/256$
$\phi_{13/49}$	$33\pi/64$
$\phi_{9/44}$	$217\pi/128$
$\phi_{10/46}$	$115\pi/128$
$\phi_{20/55}$	$963\pi/512$
$\phi_{12/48}$	$971\pi/512$
$\phi_{23/57}$	$617\pi/512$
$\phi_{15/50}$	$973\pi/512$
$\phi_{14/51}$	$701\pi/512$
$\phi_{16/52}$	$399\pi/256$
$\phi_{17/53}$	$79\pi/64$
$\phi_{18/59}$	$359\pi/256$
$\phi_{19/54}$	$1007\pi/512$
$\phi_{27/60}$	$719\pi/512$
$\phi_{22/61}$	$211\pi/128$
$\phi_{21/56}$	$319\pi/256$
$\phi_{25/63}$	$125\pi/256$
$\phi_{24/58}$	$121\pi/256$

$\tau = 1$	
$\phi_{11/47}$	$71\pi/128$
$\phi_{6/40}$	$3\pi/2$
$\phi_{13/49}$	$51\pi/32$
$\phi_{9/44}$	$\pi/2$
$\phi_{10/46}$	$755\pi/512$
$\phi_{20/55}$	$759\pi/512$
$\phi_{12/48}$	$863\pi/512$
$\phi_{23/57}$	π
$\phi_{15/50}$	$751\pi/512$
$\phi_{14/51}$	$491\pi/512$
$\phi_{16/52}$	$777\pi/512$
$\phi_{17/53}$	$581\pi/512$
$\phi_{18/59}$	$87\pi/128$
$\phi_{19/54}$	$205\pi/256$
$\phi_{27/60}$	$1015\pi/512$
$\phi_{22/61}$	$505\pi/512$
$\phi_{21/56}$	$177\pi/512$
$\phi_{25/63}$	$777\pi/512$
$\phi_{24/58}$	$317\pi/256$

$\tau = 2$	
$\phi_{11/47}$	$527\pi/512$
$\phi_{6/40}$	$133\pi/256$
$\phi_{13/49}$	$769\pi/512$
$\phi_{9/44}$	$21\pi/32$
$\phi_{10/46}$	$235\pi/512$
$\phi_{20/55}$	$931\pi/512$
$\phi_{12/48}$	$769\pi/512$
$\phi_{23/57}$	$151\pi/128$
$\phi_{15/50}$	$887\pi/512$
$\phi_{14/51}$	$\pi/16$
$\phi_{16/52}$	$51\pi/128$
$\phi_{17/53}$	$89\pi/128$
$\phi_{18/59}$	$859\pi/512$
$\phi_{19/54}$	$455\pi/256$
$\phi_{27/60}$	$175\pi/512$
$\phi_{22/61}$	$31\pi/256$
$\phi_{21/56}$	$511\pi/256$
$\phi_{25/63}$	$513\pi/512$
$\phi_{24/58}$	$257\pi/512$

$\tau = 3$	
$\phi_{11/47}$	$69\pi/128$
$\phi_{6/40}$	$461\pi/512$
$\phi_{13/49}$	$641\pi/512$
$\phi_{9/44}$	$\pi/64$
$\phi_{10/46}$	$253\pi/256$
$\phi_{20/55}$	$391\pi/256$
$\phi_{12/48}$	$279\pi/256$
$\phi_{23/57}$	$257\pi/512$
$\phi_{15/50}$	$7\pi/256$
$\phi_{14/51}$	$1013\pi/512$
$\phi_{16/52}$	$387\pi/256$
$\phi_{17/53}$	$959\pi/512$
$\phi_{18/59}$	$249\pi/256$
$\phi_{19/54}$	$795\pi/512$
$\phi_{27/60}$	$31\pi/64$
$\phi_{22/61}$	$641\pi/512$
$\phi_{21/56}$	$267\pi/512$
$\phi_{25/63}$	$59\pi/128$
$\phi_{24/58}$	$181\pi/128$

$\tau = 4$	
$\phi_{11/47}$	$461\pi/512$
$\phi_{6/40}$	$31\pi/16$
$\phi_{13/49}$	$391\pi/256$
$\phi_{9/44}$	$333\pi/512$
$\phi_{10/46}$	$97\pi/64$
$\phi_{20/55}$	$397\pi/512$
$\phi_{12/48}$	$347\pi/256$
$\phi_{23/57}$	$927\pi/512$
$\phi_{15/50}$	$487\pi/512$
$\phi_{14/51}$	$479\pi/512$
$\phi_{16/52}$	$23\pi/64$
$\phi_{17/53}$	$107\pi/64$
$\phi_{18/59}$	$401\pi/256$
$\phi_{19/54}$	$51\pi/32$
$\phi_{27/60}$	$89\pi/128$
$\phi_{22/61}$	$47\pi/32$
$\phi_{21/56}$	$113\pi/64$
$\phi_{25/63}$	$453\pi/512$
$\phi_{24/58}$	$439\pi/512$

$\tau = 5$	
$\phi_{11/47}$	$79\pi/128$
$\phi_{6/40}$	$941\pi/512$
$\phi_{13/49}$	$257\pi/512$
$\phi_{9/44}$	$419\pi/256$
$\phi_{10/46}$	$113\pi/64$
$\phi_{20/55}$	$231\pi/256$
$\phi_{12/48}$	$3\pi/2$
$\phi_{23/57}$	$439\pi/512$
$\phi_{15/50}$	$73\pi/64$
$\phi_{14/51}$	$673\pi/512$
$\phi_{16/52}$	$297\pi/256$
$\phi_{17/53}$	$667\pi/512$
$\phi_{18/59}$	$97\pi/256$
$\phi_{19/54}$	$419\pi/512$
$\phi_{27/60}$	$211\pi/512$
$\phi_{22/61}$	$897\pi/512$
$\phi_{21/56}$	$87\pi/512$
$\phi_{25/63}$	$513\pi/512$
$\phi_{24/58}$	0

FIG. 4. The three-qubit BEQS (see Algorithm 3) for $w = 012345$ and $M_{\text{POVM}} = \{|0\rangle\langle 0| + |1\rangle\langle 1|, |2\rangle\langle 2|, |3\rangle\langle 3|, |4\rangle\langle 4|, |5\rangle\langle 5|\}$; thus, $N = 6$ and $\tilde{N} = 5$. Gray nodes indicate blind oracles controlled by Oscar; white nodes indicate Alice's computation. Here the input nodes are Alice's first layer ($I = \{1, 2\}$) with input zeros, and the output nodes are Alice's last layer ($O = \{92, 93, 94\}$). The measurement angles, which are specified to 10 bits, for each node are shown in the table; the measurement order is indicated with the node numbers. This computation has success probabilities $p_s \geqslant 1 - 10^{-4}$ for all queries $\tau \in w$.

$|0\rangle\langle 0| + e^{i\alpha}|1\rangle\langle 1|$. Then we transform the result into a graph state in Fig. 4, which is runnable within the BOQC model, whose measurement angles, along with the α parameters, are the optimization parameters. Our transformation follows Ref. [33]; a few examples of such a transformation are shown in Fig. 5. In our optimization, we set $\delta = 10^{-4}$ for both decompositions, resulting in precisions $\varepsilon < 2.1\times10^{-9}$ for the gate model and $\varepsilon < 1.2\times10^{-10}$ for the BOQC model.

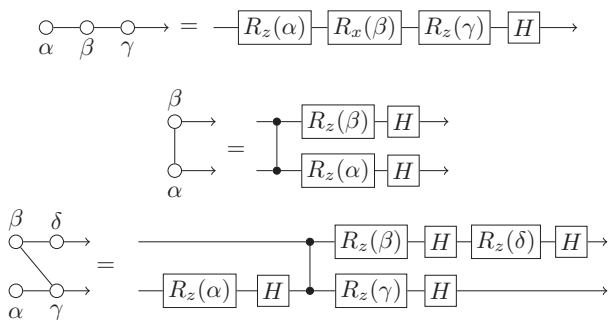
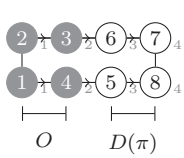


FIG. 5. Equivalent comparison between the gate model and 1WQC computations. The angles below nodes denote measurement angles; measurements are performed from left to right. The right-hand side of each graph denotes the equivalent gate model computation, assuming all measurement outcomes zero.

We have demonstrated that BEQS obtains exact quantum search algorithms with blind oracles for two computation models. Moreover, BEQS has reduced the size of computation for a five-entry database ($\tilde{N} = 5$) from using 19 CNOT gates (see Table I) to 17 CNOT gates (see Fig. 3). Our work establishes the unfortunate fact that the implementation complexity grows very rapidly for the Grover algorithm in the BOQC model. As a comparison, we obtain a cluster state for the two-qubit Grover algorithm in Fig. 6, where $\tilde{N} = 4$ and $w = 0123$. Going from a four-element database to a five-element



parameter	angle
ϕ_1	0
ϕ_2	0
ϕ_5	0
ϕ_6	0
ϕ_8	π
ϕ_7	π

$\tau = 0$		$\tau = 1$	
ϕ_3	π	ϕ_3	π
ϕ_4	π	ϕ_4	0
$\tau = 2$		$\tau = 3$	
ϕ_3	0	ϕ_3	0
ϕ_4	π	ϕ_4	0

FIG. 6. The two-qubit Grover algorithm within the BOQC scheme, with $N = \tilde{N} = 4$, $w = 0123$, and $M_{\text{POVM}} = \{|0\rangle\langle 0|, |1\rangle\langle 1|, |2\rangle\langle 2|, |3\rangle\langle 3|\}$; for notations, follow Fig. 4. Here the input nodes are $I = \{1, 2\}$ with implicit input zeros, and the output nodes are $O = \{7, 8\}$.

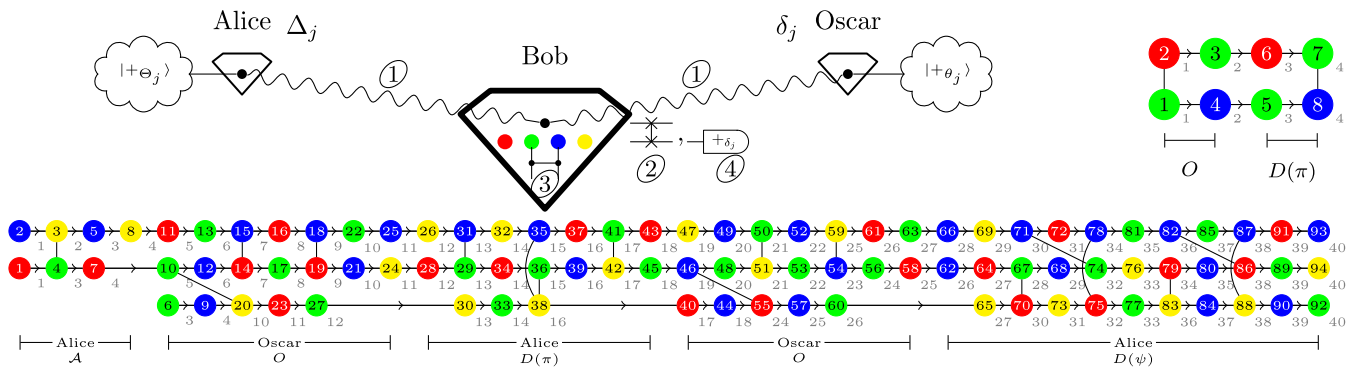


FIG. 7. Optimized BOQC protocol using three NV-center nodes; implementations of three-qubit BEQS (the 94-node graph) and two-qubit BEQS (the 8-node graph) are presented. Within the diamonds, the black node represents an electron spin and other colors represent nuclear spins; the node in the graph state will be assigned to the nuclear spin that has the same color. Bob uses his electron spin for multiple purposes: (1) as an interface to create a quantum channel with the clients, (2) as a medium to perform CPHASE gates between nuclear spins, and (3) as an ancilla to measure his nuclear spins. Alice and Oscar alternately control the computation, which comprises n rounds of steps 1–4 with the ordering shown as node numbers, where n is the number of nodes in the graph. The following sequence describes a round of Alice's moves. 1 RSP: Alice and Bob perform a heralded entanglement [44,45] to share a singlet state $|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$ between their two electron spins. Alice has in mind that she wishes to deliver state $|+_{\Theta_j}\rangle$ to Bob. She accomplishes this by measuring her electron spin (i.e., her half of the singlet state) in Θ_j; depending on her measurement outcome s_j, Bob receives $\frac{1}{\sqrt{2}}(|0\rangle + e^{i(\Theta_j + s_j\pi)}|1\rangle)$, where he knows neither Θ_j nor s_j. 2 Bob swaps the electron spin with the nuclear spin according to the color. Bob buffers all the required qubits by repeating steps 1–2, which can also be from Oscar. 3 Bob applies CPHASE gates—connecting the nodes in the graph—according to the subgraph; he connects only the nearest neighbor of the node that he is going to measure. 4 Bob measures qubit j in Δ_j—as instructed by Alice—and then announces measurement outcome b_j using a public broadcast channel. The angle Δ_j is computed by Alice taking account all the corrections. The same procedure applies to Oscar for his computations. This process is repeated until all nodes are measured. It is worth mentioning that the gray numbers represent a partial ordering induced by flow, which was computed using an algorithm of Ref. [46]. Indeed, the total ordering may be selected arbitrarily as long as the partial ordering is respected.

database for an exact quantum search algorithm within the BOQC scheme, means going from an eight-node to a 94-node cluster state.

IV. NV CENTER IMPLEMENTATION

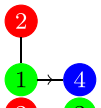
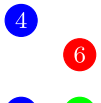
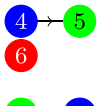
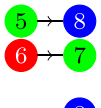
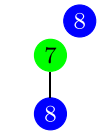
Here we introduce our proposal to implement a BOQC computation using NV centers. We propose a direct realization of the results shown above: a physical implementation of three-qubit BEQS (Fig. 4) and of the two-qubit Grover algorithm (Fig. 6). The main challenges for physical implementation are the sizable physical resources—we need 94 qubits to run three-qubit BEQS—and the high-fidelity transmission of encrypted qubits from Alice or Oscar to Bob. We think that these challenges can be at least largely overcome: To deal with the large size, we note the possibility of "reusing" the qubits [11,41] (see Sec. II B for the implementation). To accomplish reliable transmission, we propose using remote state preparation (RSP) [42] as a quantum channel. The reuse strategy drastically decreases the number of qubits: from 94 to four qubits for three-qubit BEQS and from eight to three qubits for two-qubit Grover. Moreover, RSP is understood to be very efficient for the family of states to be transmitted [43]; for RSP in our setting no additional classical communication at all is needed, automatically maintaining the blindness of the scheme.

For the BOQC implementations that we propose, the total graphs together with measurement angles are shown in Fig. 4 for three-qubit BEQS and Fig. 6 for the two-qubit Grover algorithm. Given that Alice and Oscar have successfully shared the key $\vec{r}$, and all parties agreed upon a total ordering and graphs, they thus run the protocol shown in Fig. 7. In this implementation, the BOQC protocol (see Fig. 1) is optimized to perform the computations per partition. Moreover, every partition is divided into smaller subgraphs with size four (three for two-qubit Grover). This strategy assumes that measured qubits can be reset and reused. For clarity, we tabulate in Table II the processing at each step in the BOQC (subgraphs, transmitted qubits, entanglement, and measurement angles) of the two-qubit Grover algorithm using the notation used in Sec. II B. Processing a computation of a subgraph is described as steps *1–4* in Fig. 7. Compared to the UBQC scheme, an additional set of corrections $\vec{s}$ appears as a result from the RSP.

Based on state-of-the-art technologies [23,47,48], we estimate the total computation time for three-qubit BEQS to be 3 s, where the time for each step is $t_1 \approx 25$ ms, $t_2 \approx 1.5\ \mu$s, $t_3 \approx 3.5$ ms, $t_4 \approx 0.5$ ms; assuming every two-bit operation requires 0.5 ms, and all the operations on the nuclear spins are performed by coupling the electron spin. For instance, given nuclear spins $\{n_1, n_2\}$ and electron spin e, one applies the CPHASE gate between n_1 and n_2 as follows: SWAP(n_1, e)-CPHASE(e, n_2)-SWAP(n_1, e), where each SWAP gate is implemented with three CNOT gates. For the two-qubit Grover algorithm, the total run time is estimated to be 244 ms.

Processing one subgraph, that is, executing steps *1–4*, requires one or two heralded entanglements and one or two CPHASE operations, which, on average, is completed within 31 ms for both computations. After measuring a qubit, some qubits are idle until being measured. For the three-qubit BEQS, the idling qubits need to maintain their coherence for

TABLE II. The steps on running two-qubit Grover for $\tau = 0$; that is, in this example, Alice will find database value 0 (see Fig. 6) within the BOQC scheme. On each step (i), Alice or Bob transmits fresh qubits $A(i)$ [Eq. (1)], then Bob creates the corresponding subgraph by applying entangling operator $E^{>}_{iN_{\mathcal{G}}(i)}$ [Eq. (2)]. Finally, Alice or Oscar computes the corrected measurement angle ϕ_i' from ϕ_i depending on some measurement outcomes (before Alice or Oscar encrypts ϕ_i' and sends it to Bob).

Step (i)	Subgraph	$A(i)$	$E^{>}_{iN_{\mathcal{G}(i)}}$	ϕ_i	ϕ_i'
1		{1,2,4}	$E_{12}E_{14}$	0	0
2		{3}	E_{23}	0	0
3		{6}	E_{36}	π	$(-1)^{s_2}\pi$
4		{5}	E_{45}	π	$(-1)^{s_1}\pi$
5		{8}	E_{58}	0	$s_1\pi$
6		{7}	E_{67}	0	$s_2\pi$
7		∅	E_{78}	π	$(-1)^{s_6}\pi + s_3\pi$
8		∅	–	π	$(-1)^{s_5}\pi + s_4\pi$

around 121 ms, while around one RSP and other operations are performed on the electron spin. The worst idle case happens at node 20 and 55, with idle time 377 ms while 10 RSPs are performed on the electron spin. For the two-qubit Grover, the idle average is 91 ms, while around one RSP and other operations are performed on the electron spin; the worst idle happens at node 4, which is 178 ms, while four RSPs are performed on the electron spin. We observe that the ordering we have shown here is not the optimum strategy from the point of view of the idling. It would be possible to insert some redundant nodes to reduce the idle time.

Nuclear spins on NV centers have been reported to possess coherence time of more than 1 s, even at room temperature [49,50]. Moreover, high-fidelity one- and two-bit gates operation on the nuclear spins have been demonstrated [51]. While the coherence time seems sufficient, the activities on the electron spins can decohere the idle nuclear spins, especially during the heralded entanglement attempts. Moreover, our time estimation uses the best rate known of heralded entanglement, with the nodes separated by 2 m [23]. However, the obtained heralded-entanglement fidelity is still low, hence a distillation scheme would be needed. We conclude that the current technology is still not yet ready to implement the algorithms in Fig. 7, but foreseeable improvements would make it possible.

V. CONCLUSION

We sought three-qubit and two-qubit exact quantum search algorithms within the blind oracular quantum computation scheme. Increasing the four-element database (two-qubit) into a five-element database (three-qubit) grows physical qubit requirements: from eight qubits to 94 qubits. Given that the qubits possess permanence and can be rapidly reinitialized—like NV-center qubits, the physical qubits requirement reduced to four qubits for three-qubit exact Grover and three qubits for two-qubit exact Grover. Finally, we provided an explicit implementation of the algorithms and the BOQC scheme on an NV-center platform. We estimate the coherence time required on the qubits to run such computations.

ACKNOWLEDGMENTS

We thank Tim Taminiau and Slava Dobrovitski for the insightful discussions about NV centers, Tal Mor and Rotem Liss for the discussions about distributed and blind computations, and Barbara Terhals's group in Delft and IQI members in Aachen for various discussions. We acknowledge the support of Forschungzentrum Jülich for the access to JURECA and RWTH Aachen for the access to clusters. D.D.V. acknowledges funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy–Cluster of Excellence Matter and Light for Quantum Computing (ML4Q) EXC 2004/1–390534769.

APPENDIX

1. An explicit BOQC protocol

The following protocol shows a BOQC protocol with classical inputs and classical outputs, where Bob's quantum computer is a solid state system whose qubits possess permanence and can be rapidly reinitialized. In Ref. [11], such a protocol is called BOQCo (BOQC-optimized) with classical inputs and outputs. Moreover, here we set inputs to be zeros.

2. The arbitrary step of search iteration

This section supplements Sec. II C, namely, finding the angles ψ, φ of the Høyer amplitude amplification within the operator $Q(\psi, \varphi)$ [26]. Whereas in the Grover algorithm one iteration is restricted to the rotation by $2\theta_0$, the Høyer amplitude amplification allows a rotation within the range $[-2\theta_0, 2\theta_0]$, where θ_0 is the initial angle.

Suppose that we employ n qubits and start with an equal superposition of N basis states $|\Psi_{\text{init}}\rangle$ where $2^{n-1} \leqslant N \leqslant 2^n$. Let x be the indices that can be realized by n qubits, $x = \{0, \ldots, 2^n - 1\}$ and W be a set of all possible subsets of the N-element database, $D = \{w \subseteq x : \|w\| = N\}$, and let $w \in W$, then

$$|\Psi_{\text{init}}\rangle = \frac{1}{\sqrt{N}} \sum_{j \in w} |j\rangle. \tag{A1}$$

Assume that we have an oracle that implements some function f that can distinguish weather a state is the target. Let y be the set of targets, the action of f be

$$f(j) = \begin{cases} 1, & \text{if } j \in y \\ 0, & \text{if } j \in x \setminus y. \end{cases} \tag{A2}$$

Protocol 1. [11]BOQC-optimized with classical input-output

Alice's input: $\{(\mathcal{G}, I, O), f, \vec{\Phi}\}$ ▷ $\rho_{\mathcal{A}}^{in} = \prod_{i=1}^{n} |0\rangle\langle 0|$
Oscar's input: $\{\vec{\phi}\}$
Alice's output for an honest Bob: $\rho_{\mathcal{A}}^{out} = \mathcal{E}(\rho_{\mathcal{A}}^{in})$ ▷ $\rho_{\mathcal{A}}^{out}$ is a diagonal matrix
Assumptions and conventions:
(I) Alice ($\mathcal{A}$) and Oscar ($\mathcal{O}$) have performed pre-protocol steps; Bob knows $\{(\mathcal{G}, V_{\mathcal{A}}, V_{\mathcal{O}}, \succ, >, b\}$. *Recall* $\Omega = \{\frac{\pi k}{2^{b-1}}\}_{0 \le k < 2^b}$.
(II) $invf(i) \equiv f^{-1}(i)$, $s_{invf(i)} = 0, \forall i \in I$, *and* $z(i) \coloneqq \bigoplus_{k \prec i, i \in N_{\mathcal{G}}(f(k))} s_k$.
⓪ Pre-preparation
Alice and Oscar receive a key r via a secure key channel, where $r_i \in \{0, 1\}$, for $i \in O^c$.
① BOQC by parts
for $i \in V$ with ordering $>$ **do**
  for $k \in A(i)$ **do** ▷ see Equation (1)
    if $k \in V_{\mathcal{A}}$ **then**
      Alice prepares $|+_{\Theta_k}\rangle_k$ and sends it to Bob, $\Theta_k \in \Omega$ is chosen at random.
    else if $k \in V_{\mathcal{O}}$ **then**
      Oscar prepares $|+_{\theta_k}\rangle_k$ and send it to Bob, $\theta_k \in \Omega$ is chosen at random.
    end if
  end for
  Bob applies entangling operation $E^{>}_{iN_{\mathcal{G}}(i)}$. ▷ See Equation (2)
  if $i \in V_{\mathcal{A}}$ **then**
    Alice computes $\Phi_i' = (-1)^{s_{invf(i)}}\Phi_i + z(i)\pi$.
    Alice computes $\Delta_i \coloneqq \Phi_i' + \pi r_i + \Theta_i$ and sends Bob Δ_i.
    Bob measures i in $|\pm_{\Delta_i}\rangle$ basis.
  else if $i \in V_{\mathcal{O}}$ **then**
    Oscar computes $\psi_i' = (-1)^{s_{invf(i)}}\phi_i + z(i)\pi$.
    Oscar computes $\delta_i \coloneqq \phi_i' + \pi r_i + \theta_i$ and sends Bob δ_i.
    Bob measures i in $|\pm_{\delta_i}\rangle$ basis.
  end if
  Bob sends Alice and Oscar the measurement outcome $\tilde{s}_i$.
  Alice and Oscar set $s_i = \tilde{s}_i \oplus r_i$.
end for

The function f induces a subspace spanned by "good state" $|\Psi_1\rangle = \frac{1}{\sqrt{N}} \sum_{\{j:f(j)=1\}} |j\rangle$ and "bad state" $|\Psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{\{j:f(j)=0\}} |j\rangle$. Thus, the initial state can be rewritten as $|\Psi_{\text{init}}\rangle = |\Psi_1\rangle + |\Psi_0\rangle$. Let us search for M targets. In the normalized basis of good and bad states, we rewrite again the initial state

$$|\Psi_{\text{init}}\rangle = \sqrt{a}|\tilde{\Psi}_1\rangle + \sqrt{1-a}|\tilde{\Psi}_0\rangle, \tag{A3}$$

where $|\tilde{\Psi}_1\rangle = \frac{1}{\sqrt{M}}|\Psi_1\rangle$, $|\tilde{\Psi}_0\rangle = \frac{1}{\sqrt{N-M}}|\Psi_0\rangle$, and $a = \frac{M}{N} \equiv \sin^2(\theta_0)$.

Let $Q(\varphi, \psi)$ be the operator that performs search iteration with parameters $\varphi, \psi \in [0, 2\pi)$

$$Q(\varphi, \psi) \equiv -\mathcal{A}S_0(\psi)\mathcal{A}S_y(\varphi), \tag{A4}$$

where $\mathcal{A}$ is the preparation operator that transforms state $|0\rangle^{\otimes n}$ into the equal superposition state $\mathcal{A}|0\rangle^{\otimes n} = |\Psi_{\text{init}}\rangle$. In the Grover algorithm of database size 2^n, $\mathcal{A}$ basically consists of Hadamards. Note that we can prepare $|\Psi_{\text{init}}\rangle$ from any convenient starting state. For simplicity, we start with zero state $|0\rangle^{\otimes n}$.

Essentially, $Q(\varphi, \psi)$ consists of one oracle call $S_y(\varphi)$ and a diffusion operator $D(\psi) \equiv \mathcal{A}S_0(\psi)\mathcal{A}$. The oracle call $\mathcal{S}_y(\varphi)$ "marks" the targets y by $e^{i\varphi}$, and it can be defined as

$$S_y(\varphi) \coloneqq I - (1 - e^{i\varphi})|\tilde{\Psi}_1\rangle\langle\tilde{\Psi}_1|. \tag{A5}$$

The operator $S_0(\psi)$ marks the state before preparation (in our case was $|0\rangle^{\otimes n}$) with phase $e^{i\psi}$. Thus, the diffusion operator follows

$$\begin{aligned} D(\psi) &= \mathcal{A}[I - (1 - e^{i\psi})(|0\rangle\langle 0|)^{\otimes n}]\mathcal{A} \\ &= I - (1 - e^{i\psi})|\Psi_{\text{init}}\rangle\langle\Psi_{\text{init}}|. \end{aligned} \tag{A6}$$

By using basis $\{|\tilde{\Psi}_0\rangle, |\tilde{\Psi}_1\rangle\}$, we can represent Q in matrix form

$$Q(\varphi, \psi) = \begin{bmatrix} -a(1 - e^{i\psi}) - e^{i\psi} & (1 - e^{i\psi})e^{i\varphi}\sqrt{a(1-a)} \\ (1 - e^{i\psi})\sqrt{a(1-a)} & a(1 - e^{i\psi})e^{i\varphi} - e^{i\varphi} \end{bmatrix}. \tag{A7}$$

Now, the question is: How can one implement an arbitrary rotation θ from $|\Psi_{\text{init}}\rangle$ by applying $Q(\varphi, \psi)$? We need to find what φ and ψ are, given θ. By imposing some conditions on φ and ψ, we can find them by using some tricks.

Note that we are working only with two-dimensional Hilbert space spanned by the complex vectors $\{|\tilde{\Psi}_0\rangle, |\tilde{\Psi}_1\rangle\}$. Therefore, we may associate Q with some general form of two-dimensional unitary operator.

Given an arbitrary unitary operator U with four parameters $\delta, \varphi_1, \varphi_2, \theta \in [0, 2\pi)$,

$$U = e^{i\frac{\delta}{2}} \begin{pmatrix} e^{i\varphi_1}\cos(\theta) & e^{i\varphi_2}\sin(\theta) \\ -e^{-i\varphi_2}\sin(\theta) & e^{-i\varphi_1}\cos(\theta) \end{pmatrix}. \tag{A8}$$

Let us transform the parameters into the following. Let $\varphi_1 = \mu + \nu$ and $\varphi_2 = \mu - \nu + \pi$, thus

$$U = e^{i\frac{\delta}{2}} \begin{pmatrix} e^{i\mu+i\nu}\cos(\theta) & -e^{i\mu-i\nu}\sin(\theta) \\ e^{-i\mu+i\nu}\sin(\theta) & e^{-i\mu-i\nu}\cos(\theta) \end{pmatrix}. \tag{A9}$$

We impose the condition that the diagonal elements be equal, that is, fulfilled if and only if $\varphi_1 = -\varphi_1$. This implies $\varphi_1 = 0$ and thus $\nu = -\mu$. Let us call this matrix $\tilde{U}$. Later we reparameterize $\tilde{U}$ by setting $\delta/2 = v$ and $2\mu = u$, thus

$$\begin{aligned} \tilde{U} &= e^{i\frac{\delta}{2}} \begin{pmatrix} \cos(\theta) & -e^{i2\mu}\sin(\theta) \\ e^{-i2\mu}\sin(\theta) & \cos(\theta) \end{pmatrix} \\ &= e^{iv} \begin{pmatrix} \cos(\theta) & -e^{iu}\sin(\theta) \\ e^{-iu}\sin(\theta) & \cos(\theta) \end{pmatrix}. \end{aligned} \tag{A10}$$

We factorize $\tilde{U}$ in the following way:

$$\tilde{U} = e^{iv} \begin{pmatrix} 1 & 0 \\ 0 & e^{-iu} \end{pmatrix} \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & e^{iu} \end{pmatrix}. \tag{A11}$$

In this form, it is easy to see that $\tilde{U}$ performs a real rotation up to some conditional phases. The aim is to associate our search operator Q with $\tilde{U}$.

We set Q such that its diagonal elements are also equal, which means $-a(1-e^{i\psi}) - e^{i\psi} = a(1-e^{i\psi})e^{i\varphi} - e^{i\varphi}$. From the Høyer's result [26], suppose $\varphi \neq \pi$; then this condition is fulfilled if and only if

$$\tan(\varphi/2) = \tan(\psi/2)(1-2a). \tag{A12}$$

Given $\tilde{Q}$, which is the matrix Q with equal diagonal elements, at this point, it is straightforward to parameterize $\tilde{Q}$. We can find parameters ψ and φ in the following manner:

$$\|(1-e^{i\psi})\sqrt{a(1-a)}\| = \sin(\theta),$$

$$\psi = \arccos\left(1 - \frac{\sin^2(\theta)}{2a(1-a)}\right), \tag{A13}$$

$$\varphi = 2\arctan[\tan(\psi/2)(1-2a)]. \tag{A14}$$

Now we are able to perform an arbitrary rotation θ on a state $|\Psi_{\text{init}}\rangle$ with initial angle $\theta_0 = \arcsin(\sqrt{a})$ using $\tilde{Q}(\varphi, \psi)$, up to some conditional phases. Thus, we may relate Q and $\tilde{Q}$ by canceling its conditional phases:

$$Q = e^{-iv} \begin{pmatrix} 1 & 0 \\ 0 & e^{iu} \end{pmatrix} \tilde{Q} \begin{pmatrix} 1 & 0 \\ 0 & e^{-iu} \end{pmatrix}. \tag{A15}$$

Two additional parameters are necessary in order to have a correct rotation, thus $Q = Q(\varphi, \psi, u, v)$. By knowing ψ, the phases u and v can be obtained straightforwardly, for instance,

$$v = \arg(-a(1-e^{i\psi}) - e^{i\psi}), \tag{A16}$$

$$u = v - \arg((1-e^{i\psi})\sqrt{a(1-a)}). \tag{A17}$$

Since one rotation is limited to $\theta \in [-2\theta_0, 2\theta_0]$, we need to split it into several iterations if $\|\theta\| > \|2\theta\|$. Suppose we perform $m > 1$ iterations for which each iteration rotates $\tilde{\theta} = \theta/m$ with parameters $\tilde{u}$, $\tilde{v}$; thus

$$Q^m = e^{-im\tilde{v}} \begin{pmatrix} 1 & 0 \\ 0 & e^{i\tilde{u}} \end{pmatrix} \begin{pmatrix} \cos(\tilde{\theta}) & -\sin(\tilde{\theta}) \\ \sin(\tilde{\theta}) & \cos(\tilde{\theta}) \end{pmatrix}^m \begin{pmatrix} 1 & 0 \\ 0 & e^{-i\tilde{u}} \end{pmatrix}. \tag{A18}$$

For the identical iterations, the phase corrections need only be performed once at the beginning and end, since they will be canceled out in the intermediate stages.

3. The exhaustive search circuits

This section provides details on the numerical results. First, Table III shows the equivalences of databases for three-qubit cases, related by bit permutation and bit complementation. Then, for each partition, we pick one combination of a database and compute the success probability of the BEQS algorithm by restricting the number of CNOT gates (Table IV).

TABLE III. All possible three-qubit database combinations for $5 \leqslant N \leqslant 8$. The set is partitioned by equivalence of bit permutation and bit complementation. Each partition is placed in the same row, e.g., the five-database ($N = 5$) has three partitions, namely, three unique database combinations. Simply put, each row contains equivalent database combinations .

N	Equivalent combinations of database
5	01234, 01235, 01236, 01237, 01245, 01246, 01345, 01357, 01456, 01457, 02346, 02367, 02456, 02467, 04567, 12357, 12367, 13457, 13567, 14567, 23467, 23567, 24567, 34567
	01247, 01356, 02356, 03456, 03567, 12347, 12457, 12467
	01256, 01257, 01267, 01346, 01347, 01367, 01467, 01567, 02345, 02347, 02357, 02457, 02567, 03457, 03467, 12345, 12346, 12356, 12456, 12567, 13456, 13467, 23456, 23457
6	012345, 012346, 012357, 012367, 012456, 013457, 014567, 023467, 024567, 123567, 134567, 234567
	012567, 013467, 023457, 123456
	012347, 012356, 012457, 012467, 013456, 013567, 023456, 023567, 034567, 123457, 123467, 124567
7	0123456, 0123457, 0123467, 0123567, 0124567, 0134567, 0234567, 1234567
8	01234567

TABLE IV. The success probability of three-qubit exact Grover (expected probability is 1) for all possible unique database combinations, approximated with various size CNOT gates. In particular, the success probability of the three-qubit BEQS algorithm is obtained by approximating each unitary with multiple numbers of CNOT gates. Each operator in $\{A, O(\pi), D(\pi), O(u+\phi), D(\psi)\}$ is approximated with the corresponding number of CNOT gates (on the left side), while the rest of operators are in the exact unitary form. For example, in the first row and second column of this table means: running BEQS for database choice $w = 01234$, where operator A is approximated with zero CNOT gates—while the rest of the operators $O(\pi), D(\pi), O(u+\phi), D(\psi)$ are using the exact form—gives success probability of finding the database 0.7449. The approximation of each operator is obtained by optimization method DS94.

$N = 5, \zeta + \varphi = 1.7076, \psi = 0.4510, w = 01234$

CNOT	$\mathcal{A}$	$O(\pi)$					$D(\pi)$	$O(u+\varphi)$					$D(\psi)$
		0	1	2	3	4		0	1	2	3	4	
0	0.7449	0.3463	0.8592	0.5187	0.8197	0.4388	0.2380	0.4388	0.8428	0.4449	0.8375	1.0000	0.9604
1	0.8575	1.0000	0.8062	1.0000	0.6723	1.0000	0.3114	1.0000	0.6790	1.0000	0.7084		0.9236
2	1.0000		1.0000		1.0000		0.5350		1.0000		1.0000		0.9525
3							0.7062						0.9329
4							0.7715						0.9845
5							0.8948						0.9832
6							0.9781						0.9973
7							0.9999						0.9990
8							1.0000						0.9999
9													1.0000

$N = 5, \zeta + \varphi = 1.7076, \psi = 0.4510, w = 01247$

		0	1	2	4	7		0	1	2	4	7	
0	0.6500	1.0000	0.8006	0.5706	0.9271	0.5950	0.2233	0.5950	0.9507	0.6047	0.9005	0.6313	0.9446
1	0.6581		0.9647	1.0000	0.7872	1.0000	0.1646	1.0000	0.7268	1.0000	0.7588	1.0000	0.3315
2	1.0000		1.0000		1.0000		0.3310		1.0000		1.0000		0.9543
3							0.4529						0.8952
4							0.6318						0.9766
5							0.7458						0.9731
6							0.9251						0.9912
7							1.0000						0.9979
8													0.9999

$N = 5, \zeta + \varphi = 1.7076, \psi = 0.4510, w = 01256$

		0	1	2	5	6		0	1	2	5	6	
0	0.6542	1.0000	1.0000	0.3625	0.8350	0.3108	0.2221	0.3108	0.8432	0.4082	0.8533	0.2448	0.9604
1	0.8575			1.0000	0.7236	1.0000	0.2538	1.0000	0.6802	1.0000	0.6979	1.0000	0.9279
2	0.9397				1.0000		0.3660		1.0000		1.0000		0.9717
3	1.0000						0.5617						0.8967
4	0.8184						0.7715						0.9778
5	0.9326						0.8026						0.9835
6	0.9482						0.9293						0.9979
7	1.0000						0.9996						0.9986
8							1.0000						0.9998
9													0.9999

$N = 6, \zeta + \varphi = 1.8605, \psi = 0.8411, w = 012345$

		0	1	2	3	4	5		0	1	2	3	4	5	
0	0.8024	0.3570	0.9079	0.3721	0.7899	0.4035	0.7407	0.2733	0.7407	0.3683	0.8720	0.3966	0.8621	0.2421	0.9248
1	1.0000	0.3702	0.6940	0.4278	0.7054	1.0000	0.6414	0.1892	0.6414	1.0000	0.6663	1.0000	0.6932	1.0000	0.8756
2		1.0000	0.6434	1.0000	0.6617		1.0000	0.6892	1.0000		1.0000		1.0000		0.9077
3			0.7156		0.7156			0.8268							0.8481
4			1.0000		1.0000			1.0000							0.9698
5								1.0000							0.9806
6															1.0000

TABLE IV. (*Continued.*)

$N = 6, \zeta + \varphi = 1.8605, \psi = 0.8411, w = 012347$

		0	1	2	3	4	7		0	1	2	3	4	7	
0	0.7714	0.3917	0.6254	0.3942	0.8433	0.3907	0.8684	0.2694	0.8684	0.4236	0.6345	0.4249	0.8374	0.4950	0.8914
1	0.8024	0.5393	0.6601	1.0000	0.8211	1.0000	0.7526	0.3086	0.7526	0.5410	0.6788	1.0000	0.6773	1.0000	0.1893
2	1.0000	1.0000	0.5921		1.0000		1.0000	0.5476	1.0000	1.0000	0.5958		1.0000		0.9077
3			0.7175					0.6324			0.7258				0.8295
4			1.0000					0.7424			1.0000				0.9641
5								0.8726							0.9641
6								1.0000							0.9824
7															1.0000

$N = 6, \zeta + \varphi = 1.8605, \psi = 0.8411, w = 012567$

		0	1	2	5	6	7		0	1	2	5	6	7	
0	0.7500	0.2580	0.7261	0.2841	0.7444	0.4075	0.7368	0.1925	0.7368	0.2322	0.7099	0.3807	0.7493	0.3878	0.9251
1	0.7714	1.0000	0.7025	1.0000	0.7219	1.0000	0.7394	0.3731	0.7394	1.0000	0.7227	1.0000	0.7042	1.0000	0.4771
2	0.8293		1.0000		1.0000		1.0000	0.2278	1.0000		1.0000		1.0000		0.8899
3	1.0000							0.2181							0.9037
4								0.6049							0.9617
5								0.6229							0.9631
6								0.8257							0.9813
7								0.9596							0.9996
8								1.0000							0.9999

$N = 7, \zeta + \varphi = 2.0277, \psi = 1.2056, w = 0123456$

		0	1	2	3	4	5	6		0	1	2	3	4	5	6	
0	0.8836	0.5210	0.5794	0.0810	0.4744	0.1625	0.7817	0.4387	0.2131	0.4387	0.6534	0.4318	0.6674	0.4507	0.6005	0.3290	0.8685
1	0.8890	0.4903	0.3747	0.4384	0.4728	0.3949	0.4527	1.0000	0.4133	1.0000	0.6564	0.4652	0.4527	1.0000	0.6774	1.0000	0.4798
2	0.9149	0.5569	0.4636	1.0000	0.6999	1.0000	0.7098		0.5111		1.0000	1.0000	0.6985		1.0000		0.8905
3	1.0000	1.0000	0.6879		0.7345		0.7156		0.6613				0.7136				0.8905
4			1.0000		1.0000		0.7825		0.7016				1.0000				0.9282
5							0.9111		0.7621								0.8409
6							1.0000		0.8239								0.9754
7									0.9719								0.9690
8									1.0000								0.9762
9																	1.0000

$N = 8, \zeta + \varphi = 2.2143, \psi = 1.5708, w = 01234567$

		0	1	2	3	4	5	6	7		0	1	2	3	4	5	6	7	
0	1.0000	0.6193	0.7487	0.3517	0.2817	0.3698	0.7262	0.5016	0.3434	0.0156	0.3434	0.1804	0.6695	0.5933	0.3568	0.3061	0.2336	0.5389	0.8565
1		0.6296	0.3980	0.1596	0.3178	0.6412	0.3159	0.0864	0.4831	0.7656	0.4831	0.6222	0.5543	0.5685	0.3384	0.1113	0.4132	0.6278	0.6171
2		0.5042	0.5089	0.3309	0.5023	0.3055	0.5090	0.6350	0.3895	0.5312	0.3895	0.4382	0.3703	0.6321	0.3730	0.6548	0.4884	0.1716	0.8902
3		0.6278	0.7276	0.5495	0.7781	0.5509	0.6162	0.6740	0.7633	0.5312	0.7633	0.5554	0.6200	0.4793	0.6200	0.4275	0.6542	0.7253	0.8902
4		0.5383	0.6945	0.6470	0.7831	0.5704	0.6847	0.4877	0.8043	0.8902	0.8043	0.4531	0.5539	0.6574	0.6994	0.7421	0.6840	0.5769	0.9619
5		0.7407	0.7961	0.7369	0.8875	0.7333	0.8747	0.7372	0.8763	0.8902	0.8763	0.7409	0.8778	0.7286	0.8762	0.7159	0.8760	0.7217	0.9619
6		1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

[1] C. H. Bennett and G. Brassard, Quantum cryptography: Public key distribution and coin tossing, in *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing, Bangalore, India* (IEEE, New York, 1984).

[2] S. Wiesner, Conjugate coding, SIGACT News **15**, 78 (1983).

[3] R. Cleve, W. Van Dam, M. Nielsen, and A. Tapp, Quantum entanglement and the communication complexity of the inner product function, in *Quantum Computing and Quantum Communications*, edited by C. P. Williams, Lecture Notes in Computer Science Vol. 1509 (Springer, Berlin, Heidelberg, 1999), pp. 61–74.

[4] B. Hensen, H. Bernien, A. E. Dréau, A. Reiserer, N. Kalb, M. S. Blok, J. Ruitenberg, R. F. Vermeulen, R. N. Schouten, C. Abellán *et al.*, Loophole-free Bell inequality violation using electron spins separated by 1.3 kilometres, Nature (London) **526**, 682 (2015).

[5] A. Chakrabarti, C. Lin, and N. K. Jha, Design of quantum circuits for random walk algorithms, in *Proceedings of the 2012*

IEEE Computer Society Annual Symposium on VLSI (IEEE, Amherst, MA, 2012), pp. 135–140.

[6] M. Ghosh, A. Chakrabarti, and N. K. Jha, Automated quantum circuit synthesis and cost estimation for the binary welded tree oracle, J. Emerg. Technol. Comput. Syst. **13**, 1 (2017).

[7] A. Broadbent, J. Fitzsimons, and E. Kashefi, Universal blind quantum computation, in *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, FOCS'09 (IEEE, Atlanta, GA, 2009), pp. 517–526.

[8] R. Raussendorf and H. J. Briegel, A One-Way Quantum Computer, Phys. Rev. Lett. **86**, 5188 (2001).

[9] R. Raussendorf, D. E. Browne, and H. J. Briegel, Measurement-based quantum computation on cluster states, Phys. Rev. A **68**, 022312 (2003).

[10] M. A. Nielsen, Cluster-state quantum computation, Rep. Math. Phys. **57**, 147 (2006).

[11] C. Gustiani and D. P. DiVincenzo, Blind oracular quantum computation, Quantum Sci. Technol. **6**, 045022 (2021).

[12] L. K. Grover, A fast quantum mechanical algorithm for database search, in *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing* (ACM, Philadelphia, PA, 1996), pp. 212–219.

[13] I. L. Chuang, N. Gershenfeld, and M. Kubinec, Experimental Implementation of Fast Quantum Searching, Phys. Rev. Lett. **80**, 3408 (1998).

[14] J. A. Jones, M. Mosca, and R. H. Hansen, Implementation of a quantum search algorithm on a quantum computer, Nature (London) **393**, 344 (1998).

[15] M. Anwar, D. Blazina, H. Carteret, S. Duckett, and J. Jones, Implementing Grover's quantum search on a para-hydrogen based pure state NMR quantum computer, Chem. Phys. Lett. **400**, 94 (2004).

[16] K.-A. Brickman, P. C. Haljan, P. J. Lee, M. Acton, L. Deslauriers, and C. Monroe, Implementation of Grover's quantum search algorithm in a scalable system, Phys. Rev. A **72**, 050306(R) (2005).

[17] M. Feng, Grover search with pairs of trapped ions, Phys. Rev. A **63**, 052308 (2001).

[18] L. DiCarlo, J. Chow, J. Gambetta, L. S. Bishop, B. Johnson, D. Schuster, J. Majer, A. Blais, L. Frunzio, S. Girvin *et al.*, Demonstration of two-qubit algorithms with a superconducting quantum processor, Nature (London) **460**, 240 (2009).

[19] S. Yao, A. Qing, and L. Gui-Lu, A scheme for simulation of quantum gates by Abelian anyons, Commun. Theor. Phys. **56**, 873 (2011).

[20] P. Walther, K. J. Resch, T. Rudolph, E. Schenck, H. Weinfurter, V. Vedral, M. Aspelmeyer, and A. Zeilinger, Experimental one-way quantum computing, Nature (London) **434**, 169 (2005).

[21] K. Chen, C.-M. Li, Q. Zhang, Y.-A. Chen, A. Goebel, S. Chen, A. Mair, and J.-W. Pan, Experimental Realization of One-Way Quantum Computing with Two-Photon Four-Qubit Cluster States, Phys. Rev. Lett. **99**, 120503 (2007).

[22] S. Barz, E. Kashefi, A. Broadbent, J. F. Fitzsimons, A. Zeilinger, and P. Walther, Demonstration of blind quantum computing, Science **335**, 303 (2012).

[23] P. C. Humphreys, N. Kalb, J. P. Morits, R. N. Schouten, R. F. Vermeulen, D. J. Twitchen, M. Markham, and R. Hanson, Deterministic delivery of remote entanglement on a quantum network, Nature (London) **558**, 268 (2018).

[24] Z. Diao, Exactness of the original Grover search algorithm, Phys. Rev. A **82**, 044301 (2010).

[25] D. P. Chi and J. Kim, in *Quantum Computing and Quantum Communications, First NASA International Conference, Selected Papers, QCQC'98*, edited by C. P. Williams, Lecture Notes in Computer Science Vol. 1509 (Springer-Verlag, Berlin, 1999), pp. 148–151.

[26] P. Høyer, Arbitrary phases in quantum amplitude amplification, Phys. Rev. A **62**, 052304 (2000).

[27] G.-L. Long, Grover algorithm with zero theoretical failure rate, Phys. Rev. A **64**, 022307 (2001).

[28] Y. Liu, An exact quantum search algorithm with arbitrary database, Int. J. Theor. Phys. **53**, 2571 (2014).

[29] W. L. Yang, C. Y. Chen, and M. Feng, Implementation of three-qubit Grover search in cavity quantum electrodynamics, Phys. Rev. A **76**, 054301 (2007).

[30] L. M. K. Vandersypen, M. Steffen, M. H. Sherwood, C. S. Yannoni, G. Breyta, and I. L. Chuang, Implementation of a three-quantum-bit search algorithm, Appl. Phys. Lett. **76**, 646 (2000).

[31] C. Figgatt, D. Maslov, K. Landsman, N. Linke, S. Debnath, and C. Monroe, Complete 3-qubit Grover search on a programmable quantum computer, Nat. Commun. **8**, 1918 (2017).

[32] Y. Liu and F. Zhang, First experimental demonstration of an exact quantum search algorithm in nuclear magnetic resonance system, Sci. China Phys. Mech. Astron. **58**, 1 (2015).

[33] V. Danos and E. Kashefi, Determinism in the one-way model, Phys. Rev. A **74**, 052310 (2006).

[34] C. Zalka, Grover's quantum searching algorithm is optimal, Phys. Rev. A **60**, 2746 (1999).

[35] G. L. Long, Y. S. Li, W. L. Zhang, and L. Niu, Phase matching in quantum searching, Phys. Lett. A **262**, 27 (1999).

[36] G.-L. Long, X. Li, and Y. Sun, Phase matching condition for quantum search with a generalized initial state, Phys. Lett. A **294**, 143 (2002).

[37] D. P. DiVincenzo and J. Smolin, Results on two-bit gate design for quantum computers, in *Proceedings of the Workshop on Physics and Computation*, PhysComp '94 (IEEE Press, Dallas, TX, 1994), p. 14.

[38] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, in *Numerical Recipes* (Cambridge University Press, New York, 1986), Chap. 10, pp. 426–430.

[39] G. Vidal and C. M. Dawson, Universal quantum circuit for two-qubit transformations with three controlled-NOT gates, Phys. Rev. A **69**, 010301(R) (2004).

[40] E. Jones *et al.*, SciPy 1.0: Fundamental algorithms for scientific computing in python, Nat. Meth. **17**, 261 (2020).

[41] M. Houshmand, M. Houshmand, and J. F. Fitzsimons, Minimal qubit resources for the realization of measurement-based quantum computation, Phys. Rev. A **98**, 012318 (2018).

[42] C. H. Bennett, D. P. DiVincenzo, P. W. Shor, J. A. Smolin, B. M. Terhal, and W. K. Wootters, Remote State Preparation, Phys. Rev. Lett. **87**, 077902 (2001).

[43] H.-K. Lo, Classical-communication cost in distributed quantum-information processing: A generalization of quantum-communication complexity, Phys. Rev. A **62**, 012313 (2000).

[44] S. D. Barrett and P. Kok, Efficient high-fidelity quantum computation using matter qubits and linear optics, Phys. Rev. A **71**, 060310(R) (2005).

[45] H. Bernien, B. Hensen, W. Pfaff, G. Koolstra, M. Blok, L. Robledo, T. Taminiau, M. Markham, D. Twitchen, L. Childress *et al.*, Heralded entanglement between solid-state qubits separated by three metres, Nature (London) **497**, 86 (2013).

[46] M. Mhalla and S. Perdrix, Finding optimal flows efficiently, in *International Colloquium on Automata, Languages, and Programming* (Springer, Reykjavik, Iceland, 2008), pp. 857–868.

[47] N. Kalb, A. A. Reiserer, P. C. Humphreys, J. J. Bakermans, S. J. Kamerling, N. H. Nickerson, S. C. Benjamin, D. J. Twitchen, M. Markham, and R. Hanson, Entanglement distillation between solid-state quantum network nodes, Science **356**, 928 (2017).

[48] J. Cramer, N. Kalb, M. A. Rol, B. Hensen, M. S. Blok, M. Markham, D. J. Twitchen, R. Hanson, and T. H. Taminiau, Repeated quantum error correction on a continuously encoded qubit by real-time feedback, Nat. Commun. **7**, 11526 (2016).

[49] P. C. Maurer, G. Kucsko, C. Latta, L. Jiang, N. Y. Yao, S. D. Bennett, F. Pastawski, D. Hunger, N. Chisholm, M. Markham *et al.*, Room-temperature quantum bit memory exceeding one second, Science **336**, 1283 (2012).

[50] F. Dolde, I. Jakobi, B. Naydenov, N. Zhao, S. Pezzagna, C. Trautmann, J. Meijer, P. Neumann, F. Jelezko, and J. Wrachtrup, Room-temperature entanglement between single defect spins in diamond, Nat. Phys. **9**, 139 (2013).

[51] T. H. Taminiau, J. Cramer, T. van der Sar, V. V. Dobrovitski, and R. Hanson, Universal control and error correction in multi-qubit spin registers in diamond, Nat. Nanotechnol. **9**, 171 (2014).