I. INTRODUCTION

Quantum algorithms realized by quantum computers are elaborately designed to outperform the best classical algorithms [1]. Many non-deterministic polynomial-time (NP) hard program still only have the exhaustive search way to solve [2]. One-way function (oracle) \( f(x) (f : \{0, 1\}^n \rightarrow \{0, 1\}) \) can identify the solution state: if \( t \) is solution (target state), then \( f(t) = 1 \), otherwise one way function outputs 0. Classical way to tackle the exhaustive search is by querying each state in solution space (with dimension \( N \)) by the one-way function. In worst cases, the total number of query to oracle is \( N - 1 \). The principle of quantum superposition provides new way to perform the exhaustive search. Suppose that \( N = 2^n \), where \( n \) is number of qubits to represent the database. Grover’s algorithm can find one target state with oracle complexity \( O(\sqrt{N}) \), which is quadratically outperforms the classical algorithm [3, 4]. The oracle in Grover’s algorithm is \( U_f : U_f|x⟩|y⟩ = |x⟩|f(x) \oplus y⟩ \) with \( x \in \{0, 1\}^n \) and \( y \in \{0, 1\} \).

Quantum computers have been vastly developed in last ten years [5–8]. Still shallow depth algorithm can be realized on real quantum computers: the noisy intermediate-scale quantum (NISQ) era, see [9]. Width (number of physical qubits) represents the size of quantum computers. Algorithm’s depth (number of consecutive parallel gate operations) represents the real physical implantation time for the algorithm. Combined with width and depth, quantum volume gives metric for NISQ computers [10]. Coherence time is limited in NISQ computers. Discrete elementary gate set which can approximate any unitary operations is called universal quantum gate set (Solovay-Kitaev theorem) [1]. Quantum computers are equipped with universal quantum gate set. Therefore, depth is counted from universal quantum gate operations.

Quantum oracle \( U_f \) is realized by quantum gates from universal quantum gate set. It is reasonable to assume that quantum oracle can be efficiently implemented (in order to efficiently implement Grover’s algorithm), such as the oracle depth scales polynomial with \( n \) [7]. Oracle complexity will be equivalent with depth complexity if quantum oracle is the only operations realized in Grover’s algorithm. However, it is not true. Another unitary operations called diffusion operator is required in Grover’s algorithm [3, 4]. How to choose the diffusion operator is related with initial state preparation [11, 12]. The unstructured population space \( \{0, 1\}^n \) can be prepared in equal superposition state on quantum computer:

\[
|s_n⟩ = H^\otimes n|0^\otimes n
\]

with single-qubit Hadamard gate \( H \) [1]. Note that the initial state \( |s_n⟩ \) can be efficiently prepared with depth 1 circuit. Diffusion operator may only have the constrain: state \( |s_n⟩ \) is the eigenvector with eigenvalue 1 [13, 14].

Grover’s algorithm is the only threat to post-quantum cryptography. Post-quantum cryptography standardization proposed by NIST in 2016 introduces depth bound. Recently, more studies focus on the resource estimation, such as width and depth, for Grover’s algorithm instead of traditional oracle complexity [15, 16]. Grover’s algorithm is optimal in oracle complexity [17, 18]. However, no research addresses depth optimum of Grover’s algorithm. Surprisingly, the depth of diffusion operator can be reduced to one [19, 20]. However, the algorithms will have 1/2 maximal successful probability, and the expected depth is not efficient as the original Grover’s algorithm. Inspired from quantum partial search algorithm (QPSA) [21–24], we introduce new depth optimization for quantum search algorithm. New algorithm can have lower depth compared with Grover’s algorithm. In order to further lower the depth, we can apply divided and conquer strategy (combined with depth optimization). Divided and conquer strategy means that the search algorithm is realized by several stages. Each stage can find partial address of the target state. And the initial state takes rescaled version of last stage initial state. The divided and conquer strategy naturally allows the parallel running of quantum search algorithm.

If oracle takes much more depths than diffusion operator depth, oracle complexity will be approximated equivalent with depth complexity. We can define the ratio between oracle depth and diffusion operator depth. Above a critical ratio, Grover’s algorithm is optimal in depth. Based on the depth optimization method proposed in this paper, we will show that the critical ratio is proportional to \( O(n^{-1}2^{n/2}) \). If we divide the algorithm into two stages, the critical ratio is a constant.

The paper is organized as follow. In Sec. II, we briefly review on quantum search algorithms. One is Grover’s original algorithm and the other is quantum partial search algorithm (QPSA). And we set notations. In Sec. III, we introduce the depth optimization method for quantum search algorithm. And we show how to combine the divided and conquer strat-
ergy with depth optimizations. In Sec. IV, we talk about the critical ratios. Below the critical ratios, we can have search algorithm which has lower depth compared with Grover’s algorithm. Parallel running quantum search algorithm is briefly discussed in Sec. V. Sec. VI is conclusion and outlook. We provide three Appendices. Appendix A provides detailed examples on $n = 6$ search algorithm with depth optimizations; Appendix B lists the numerical details provided in the main text; Appendix C shows the numerical values for critical ratios.

II. REVIEW ON QUANTUM SEARCH ALGORITHMS

A. Grover’s Algorithm

Quantum oracle $U_f$ flips the ancillary qubit if the target state $|t⟩$ is feed in. Ancillary qubit can be prepared in superposition state $H|1⟩ = (|0⟩ - |1⟩)/\sqrt{2}$. Then the oracle gives sign flip if acting on the target state:

$$U_f(\mathbb{I}_{2^n} \otimes H)|x⟩ \otimes |1⟩ = (-1)^{f(x)}(\mathbb{I}_{2^n} \otimes H)|x⟩ \otimes |1⟩ \tag{2}$$

Here $\mathbb{I}_{2^n}$ is identity operator on $2^n$ dimensional Hilbert space. For convenience, we denote oracle $U_f$ as

$$U_t = \mathbb{I}_{2^n} - 2|t⟩⟨t| \tag{3}$$

if the ancillary qubit $H|1⟩$ is prepared. General phase flip can be constructed: $U_{t,\phi} = \mathbb{I}_{2^n} - (1 - e^{-i\phi})|t⟩⟨t|$ with complex unit $i = \sqrt{-1}$. Generalized oracle $U_{t,\phi}$ has application in sure success search algorithm [12, 25] and fixed point search algorithm (for unknown number of target state) [26]. Note that operator $U_{t,\phi} (\phi \neq \pi)$ can be realized by two quantum oracles $U_f$ [26]. In this paper, we do not consider the generalized oracle $U_{t,\phi}$ for low depth consideration. We also concentrate on one target state case. The depth optimization method in Sec. III can be easily generalized to multi-target case.

Oracle $U_t$ reflects the state over the plane perpendicular to the target state. The most efficient diffusion operator (unstructured database search) is

$$D_n = 2|s_n⟩⟨s_n| - \mathbb{I}_{2^n} \tag{4}$$

Note that $|s_n⟩$ (1) is the equal superposition of all database. Operator $D_n$ can be viewed as reflection the amplitude in the average. Diffusion operator $D_n$ does not query oracle. Therefore the oracle complexity does not include the resource cost by $D_n$. Diffusion operator $D_n$ is single-qubit gates equivalent to generalized $n$-qubit Toffoli gate $\Lambda_{n-1}(X)$ [1]. Here $X$ is NOT gate (Pauli-X gate). The notation $\Lambda_{n-1}(X)$ implies $n - 1$ control qubits NOT gate. When $n = 3$, $\Lambda_2(X)$ is Toffoli gate. When $n = 2$, $\Lambda_1(X)$ is CNOT gate. How to realize $\Lambda_{n-1}(X)$ gate on real quantum computer is highly nontrivial. It is well-known that $n$-qubit $\Lambda_{n-1}(X)$ gate can be constructed with linear $n$-depth or quadratic linear $n^2$-depth from universal gate set (CNOT gate plus single-qubit gates) [27]. Recent works also show that $n$-qubit $\Lambda_{n-1}(X)$ can be realized in $\log n$-depth if $n$-qubit ancillary qubits are provided [28] or qutrit states are applied [29].

Note that $\Lambda_{n-1}(X)$ construct with linear $n$-depth if $n$-depth or quadratic linear $n^2$-depth from universal gate set (CNOT gate plus single-qubit gates) [27]. Recent works also show that $n$-qubit $\Lambda_{n-1}(X)$ can be realized in $\log n$-depth if $n$-qubit ancillary qubits are provided [28] or qutrit states are applied [29].

FIG. 1. Quantum circuits of global Grover operator $G_n$ defined in (5) and local Grover operator defined in (8). The diffusion operator $D_n$ ($D_m$) is single-qubit gates equivalent with $n$-qubit Toffoli gate $\Lambda_{n-1}(X)$ gate ($m$-qubit Toffoli gate $\Lambda_{m-1}(X)$) [1]. Here $X$ and $Z$ are Pauli gates. And $H$ is Hadamard gate. The subspace acted by $D_n$ can be chosen arbitrarily.

One query to oracle $U_t$ defined in (3) combined with diffusion operator $D_n$ defined in (4) is called Grover iteration or Grover operator:

$$G_n = D_n U_t \tag{5}$$

See FIG. 1a for quantum circuit diagram of $G_n$. Diffusion operator $D_n$ reflects the average of whole database. Operator $G_n$ is also called global Grover iteration (global Grover operator). One Grover operator $G_n$ uses one query to oracle $U_t$. Applying $G_n$ iteratively on initial state $|s_n⟩$, the amplitude of target state will be amplified. After $j$ Grover iteration, the success probability $P_n(j)$ is

$$P_n(j) = |⟨t|G_n^j|s_n⟩|^2 = \sin^2((2j + 1)\theta) \tag{6}$$

with $\sin \theta = 1/\sqrt{N}$. When $j$ reaches to $j_{\text{max}} = [\pi/\sqrt{N}/4]$, probability of finding the target state approaches to 1. Maximal iteration number $j_{\text{max}}$ is square root of $N$. Clearly, Grover’s algorithm provides quadratic speed up compared with classical algorithm (in oracle complexity). The idea behind Grover’s algorithm can be generalized called amplitude amplification algorithm [12].

The success probability (finding the target state) does not scale linearly with number of iterations. It suggests that Grover’s algorithm becomes less efficient when $j$ approaches to $j_{\text{max}}$. Previous works argued that the expected number of iterations $j/P_n(j)$ has the minimal at $j_{\text{exp}} = [0.583\sqrt{N}]$, which is smaller than $j_{\text{max}}$ [17, 30]. When $j = j_{\text{exp}}$, the success probability is around 0.845. In practice, iteration number $j_{\text{exp}}$ has high probability to find target state. The measurement result can be verified in classical way. If the result fails, run the algorithm again. The expected number of oracles is minimized at $j_{\text{exp}}$. 
B. Quantum Partial Search Algorithm

Quantum partial search algorithm (QPSA) was introduced by Grover and Radhakrishnan [21]. Since Grover’s algorithm is optimal (in oracle complexity), QPSA trades accuracy for speed. Database of $N$ items is divided into $K$ blocks: $N = bK$. Here $b$ is the number of items in each block. We can assume number $b$ is also power of 2: $b = 2^m$. And the number of blocks is $K = 2^{n-m}$. QPSA can find the block which has the target state. In other words, QPSA finds partial $(n-m)$-bits of target state (which is $n$-bits long). The optimized QPSA can win over Grover’s algorithm a number scaling as $\sqrt{b}$ [21–23]. Larger block size (less accuracy) gives faster algorithm.

Suppose that the address of target state $|t\rangle$ is divided into $|t\rangle = |t_1\rangle \otimes |t_2\rangle$. Here $t_1$ is $(n-m)$-bits long and $t_2$ is $m$-bits long. The task is to find $t_1$ instead of whole $t$. Besides diffusion operator $D_n$ (4), QPSA introduces a new diffusion operator $D_{n,m}$:

$$D_{n,m} = \mathbb{1}_{2n-m} \otimes (2|s_m\rangle\langle s_m| - \mathbb{1}_{2m})$$  \hspace{1cm} (7)

Diffusion operator $D_{n,m}$ reflects around the average in block-size. Diffusion operator $D_{n,m}$ can be viewed as rescaled version of $D_n$ (4): the database with size $2^n$ is rescaled into size $2^m$. We can define a new Grover operator as

$$G_{n,m} = D_{n,m}U_t$$  \hspace{1cm} (8)

See FIG. 1b for quantum circuit diagram of $G_{n,m}$. Diffusion operator $D_m$ reflects the average of block items. Operator $G_m$ is also called local Grover iteration (local Grover operator). For simplicity, we shorten the notations as $D_m \equiv D_{n,m}$ and $G_m \equiv G_{n,m}$ in the rest of paper.

QPSA is realized by iteratively applying operators $G_m$ and $G_n$ on the initial state $|s_n\rangle$. Then partial bit $t_1$ can be found with high probability (computational basis measurement on final state). In QPSA, amplitudes of all non-target items in target block are same; amplitudes of all items in non-target blocks are same. Therefore, we can follow three amplitudes only. Let us introduce basis:

$$|t\rangle = |t_1\rangle \otimes |t_2\rangle,$$  \hspace{1cm} (9a)

$$|ntt\rangle = \frac{1}{\sqrt{b-1}} \sum_{j \neq t_2} |t_1\rangle \otimes |j\rangle,$$  \hspace{1cm} (9b)

$$|u\rangle = \frac{1}{\sqrt{N-b}} \left( \sqrt{N} |s_n\rangle - |t\rangle - \sqrt{b-1}|ntt\rangle \right).$$  \hspace{1cm} (9c)

State $|ntt\rangle$ is normalized sum of all non-target state in target block. State $|u\rangle$ is normalized sum of all items in non-target blocks. At new basis, initial state $|s_n\rangle$ (1) can be rewritten as

$$|s_n\rangle = \sin \gamma \sin \theta_2 |t\rangle + \sin \gamma \cos \theta_2 |ntt\rangle + \cos \gamma |u\rangle$$  \hspace{1cm} (10)

Angle $\theta_2$ is defined as $\sin \theta_2 = 1/\sqrt{b}$. Angle $\gamma$ is defined as $\sin \gamma = 1/\sqrt{K}$. Global Grover operator $G_n$ defined in (5) and local Grover operator $G_m$, defined in (8) can be reformulated as elements in $O(3)$ group [31]. Operators $G_m$ and $G_n$ have highly non-trivial commutation relations [31]. The order is the key in QPSA. Extensive studies have suggested that the optimal sequence (in oracle complexity) is $G_n G_m^2 G_n^3$ [24, 31]. Here we can minimize the number of oracle (minimize $j_1 + j_2 + 1$) given by a threshold success probability. QPSA requires less number of oracles (the saved oracle number scales as $\sqrt{b}$) than Grover’s algorithm. QPSA can also be generalized into multi-target cases [32, 33]. Interestingly, QPSA can be performed in a hierarchy way: every time QPSA finds several bits of the target bits $t$ [34].

III. DEPTH OPTIMIZATIONS

A. Minimal Expected Depth

Depth is defined as number of consecutive parallel gate operations. For example, the initial state $|s_n\rangle$ can be ppered with one depth circuit, see (1). Suppose diffusion operator $D_n$ in (4) has depth $d(D_n)$, which is same as depth of $n$-qubit generalized Toffoli gate $\Lambda_{n-1}(X)$ [11]. Different search tasks have different oracle realizations. We denote the depth ratio between oracle $U_t$ and diffusion operator $D_n$ as $\alpha$:

$$\alpha = \frac{d(U_t)}{d(D_n)}$$  \hspace{1cm} (11)

It is an important parameter for depth optimization. For one-item search algorithm, the practical minimal value for $\alpha$ is $1$: $\alpha \geq 1$. The ratio $\alpha$ maybe different for same oracle with different $n$. We fix $n$, then ratio $\alpha$ is a constant for one problem. The design for low depth generalized Toffoli gate can be benefit for oracle depth either [29]. Given by $d(D_n)$ and $\alpha$, Grover’s algorithm can be mapped to depth complexity directly. We define the minimal expected depth (MED) of Grover’s algorithm as:

$$d_G(\alpha) = \min_j d(G_n^j)$$  \hspace{1cm} (12)

Here $P_n(j)$ defined in (6) is the success probability of finding the target state (with $j$ Grover iterations). The numerator denotes the depth $d(G_n^j) = (\alpha + 1)j d(D_n)$. Above optimization is same as the expected iteration number optimization $j/P_n(j)$ [17, 30], up to a constant factor. Therefore, we can use $j_{\exp} = [0.583\sqrt{N}]$ in MED. Note that we have $P_n(j_{\exp}) \approx 0.845$. Then we have

$$d_G(\alpha) \approx 0.69 \times 2^{n/2}(\alpha + 1)d(D_n)$$  \hspace{1cm} (13)

If the oracle can be constructed in polynomial depth $d(U_t) = O(n^k)$. Then the MED of Grover’s algorithm scales as $O(n^k 2^{n/2})$ (assume that $k > 1$). Grover’s algorithm is optimal in oracle complexity [17, 18]. The minimal expected iteration number $j_{\exp}$ is optimal. The scale $O(n^k 2^{n/2})$ is also optimal for depth complexity. However, we will show that the number $d_G(\alpha)$ in (13) is not optimal (if $\alpha$ in (11) is not infinite large).
B. Optimization Method

Local diffusion operator $D_m$ defined in (7) has lower depth than global diffusion operator $D_n$ in (4). The optimization idea is replacing global diffusion operator by local diffusion operator. The global Grover operator $G_n$ defined in (5) does not commute with local Grover operator $G_m$ in (8) [31]. Then the order of $G_n$ and $G_m$ is important. Suppose that we have the sequence

$$S_{n,m}(j_1, j_2, \ldots, j_q) = G_{n}^{j_1}G_{m}^{j_2} \cdots G_{n}^{j_q}G_{m}^{j_q}$$

Here $\{j_1, j_2, \ldots, j_q\}$ are some non-negative integers. We have total

$$j_{tot} = \sum_{p=1}^{q} j_P$$

number of query to oracles. To remove the ambiguity in notation $S_{n,m}(j_1, j_2, \ldots, j_q)$, we require that the last number $j_q$ is always the number of local Grover operators. For example, $S_{6,4}(1,2) = G_6G_4^2$ and $S_{6,4}(1,1,0) = G_4G_6$. Note that $S_{n,m}(j, 0) = G_{n}^{j}G_{m}$ is the original Grover algorithm. Since the sequence $S_{n,m}(j, 0) = G_{n}^{j}$ does not have any local Grover operators, the number $m$ is irrelevant. As convention, we choose the notation $S_{n}(j, 0) = S_{n,m}(j, 0)$. The sequence $S_{n,m}(j_1, j_2, \ldots, j_q)$ can find the target state with probability:

$$P_{n,m}(j_1, j_2, \ldots, j_q) = \frac{1}{(\sqrt{N})^q} \left| \langle \alpha | S_{n,m}(j_1, j_2, \ldots, j_q) | s_0 \rangle \right|^2$$

Then we can define the expected depth of $S_{n,m}(j_1, j_2, \ldots, j_q)$ algorithm. We want to minimize the expected depth, like for Grover’s algorithm (12). Define the new MED:

$$d_1(\alpha) = \min_{m, j_1, j_2, \ldots, j_q} \frac{d(S_{n,m}(j_1, j_2, \ldots, j_q))}{P_{n,m}(j_1, j_2, \ldots, j_q)}$$

The minimization goes through non-negative integers $\{j_1, j_2, \ldots, j_q\}$. We also optimize the number $m$ (positive integer), which is $m < n$. The minimal value for $m$ is 2. The subscript 1 defined in $d_1(\alpha)$ suggests that we find the target state at one stage, i.e., no measurement within the algorithm until the end. In quantum circuit model, one-stage algorithm means only three steps: initialization, unitary operations and measurements. We can define multi-stage algorithms, which have several rounds of initialization, unitary operations and measurements. Later we will define the MED of multi-stage search algorithms.

Let us see one example. For $n = 6$, the Grover’s algorithm has MED when $j = 4$:

$$P_6(4) = \frac{1}{(\sqrt{N})^4} \left| \langle \alpha | G_6^4 | s_0 \rangle \right|^2 \approx 0.816$$

The new sequence is

$$S_{6,4}(1, 1, 2) = G_4G_6G_4^2$$

And $S_{6,4}(1, 1, 2)$ gives the success probability

$$P_{6,4}(1, 1, 2) = \frac{1}{(\sqrt{N})^3} \left| \langle \alpha | S_{6,4}(1, 1, 2) | s_0 \rangle \right|^2 \approx 0.755$$

Note that both sequences $G_6^4$ and $G_4G_6G_4^2$ have four oracles. According to [27], 6-qubit and 4-qubit Toffoli gates can be decomposed into 64 and 16 depth circuits (with single- and two-qubit gates). We suppose that $d(D_n) = 64$ and $d(D_4) = 16$. One can find: if the ratio $\alpha$ in (11) is $\alpha < 2.029$, then new sequence $G_4G_6G_4^2$ has lower expected depth. More examples (about $n = 6$ search algorithm) with quantum circuit diagrams can be found at Appendix A.

We can go back to Grover’s algorithm if the number of $G_m$ is zero. We always have

$$d_1(\alpha) \leq d_2(\alpha)$$

The choice of subspace (acted by local diffusion operators $D_m$ defined in (7)) can be arbitrary, such as qubits with high connectivity in real quantum computers. But all local diffusion operators $D_m$ should act on the same qubits. For example, the sequence $S_{6,4}(1, 1, 2)$ have three local Grover operators. The three local diffusion operators are acting on the same 4 qubits. Making wrong choice of subspace can dramatically increase the number of invariant amplitude subspace. Such strategy may have some advantages in search algorithm, but it is beyond in this paper.

The minimization results will depend on: the size of database (number $n$); the ratio between oracle depth $d(U_t)$ and diffusion operator depth $d(D_n)$ (the value of $\alpha$ defined in (11)); how $d(D_n)$ scales with $n$ (logarithmic, linear or quadratic linear with $n$). In numerical optimizations, we can set some constraints which rule out the possibility $d_1(\alpha) < d_2(\alpha)$. For example, we can set the total number of $G_n$ is less than $\lfloor 0.69\sqrt{N} \rfloor$; if the number of $G_n$ is $j$, then the number of $G_m$ should be less than $\lfloor (0.69\sqrt{N} - j)(\alpha + 1)/\alpha \rfloor$. As examples, we find the optimal sequence for $n = 4, 5, \ldots, 10$ with $\alpha = 1$ (assuming $O(n)$ depth of $\Lambda_{n-1}(X)$ gate [27]). The estimated depths are plotted in Fig. 2. Details about the corresponding optimal sequences and success probabilities are provided in Appendix B.

C. Depth Optimizations for Multi-stage Quantum Search Algorithms

In NISQ era, errors can be suppressed if a long algorithm is divided into shorter pieces (by new initialization and measurements). Inspired by hierarchy QPSA [34], we propose depth optimizations for multi-stage quantum search algorithm. For simplicity, we consider the two-stage quantum search algorithm firstly.

Suppose that the target state is divided into two-parts:

$$|t\rangle = |t_1\rangle \otimes |t_2\rangle$$

Suppose that the bit length of $t_1$ is $m_1$ and the bit length of $t_2$ is $m_2$. Note that we have $m_1 + m_2 = n$. After first stage, the search algorithm can find $|t_1\rangle$ with high probability. Based on the result on first stage, we can rescale the database. After second stage, the algorithm can find $|t_2\rangle$ with high probability. The algorithm has the following steps:

Step 1: Initialize the state to $|s_n\rangle$ defined in (1).
FIG. 2. (a) Estimated $d_0(\alpha)$ (MED of Grover’s algorithm defined in (12)), $d_1(\alpha)$ defined in (17) and $d_2(\alpha)$ defined in (26) with $\alpha = 1$. Depth $d(D_n)$ is counted based on the optimal results in [27]. The corresponding optimal sequences and success probabilities are listed in Appendix B. (b) Depth of the optimal sequence. The left (red) bar is Grover’s algorithm. The right (green) bar is the optimal sequence from $d_1(1)$ defined in (17). Since $d_2(1)$ has two stages. The middle bottom bar is the depth of first stage circuit and the middle up bar is the depth of second stage circuit.

Step 2: Perform the sequence

$$S_{n,m_2}^{(1)}(j_1,j_2,\ldots,j_q) = G_{n,m_2}^{j_q}G_{n,m_2}^{j_{q-1}}\cdots G_{n,m_2}^{j_2}G_{n,m_2}^{j_1}$$

(23)

on the initial state $|s_n\rangle$. The local diffusion operator $D_{m_2}$ (defined in $G_{m_2}$) is acting on $m_2$ qubits.

Step 3: Measure the qubits (computational basis measurements) which does not have the local diffusion operator $D_{m_2}$ acting on. Suppose that we get the classical results: $t_1' \in \{0,1\}^{m_1}$. The probability that $t_1' = t_1$ is denoted as $P_{n,m_2}^{(1)}(j_1,j_2,\ldots,j_q)$.

Step 4: Initialize the state to

$$|t_1'\rangle \otimes |s_{m_2}\rangle$$

Here $|s_{m_2}\rangle$ is the rescaled initial state:

$$|s_{m_2}\rangle = H^{\otimes m_2}|0\rangle^{\otimes m_2}$$

(24)

Step 5: Perform the sequence

$$S_{m_2,m'}^{(2)}(j_1',j_2',\ldots,j_q') = G_{m_2}^{j_q'}G_{m_2}^{j_{q-1}'}\cdots G_{m_2}^{j_2'}G_{m_2}^{j_1'}$$

(25)

on the new initial state. We have $m' < m_2$. The diffusion operator $D_{m'}$ (defined in $G_{m_2}$) is acting on $|s_{m_2}\rangle$. And the diffusion operator $D_{m'}$ is acting on the subspace of $|s_{m_2}\rangle$.

Step 6: Measure the qubits (computational basis measurements) which have the initial state $|s_{m_2}\rangle$. Suppose that we get the classical results: $t_2' \in \{0,1\}^{m_2}$. The probability that $t_2' = t_2$ is denoted as $P_{m_2,m'}^{(2)}(j_1',j_2',\ldots,j_q')$.

Step 7: Verify the solution $|t'\rangle = |t_1'\rangle \otimes |t_2'\rangle$ by classical oracle. If the solution is the target item, then stop; if not, back to Step 1.

Step 1-3 is the first stage: we find $t_1$ with high probability. Step 4-6 is the second stage: we find the remaining bits of target state. Step 7 is to verify. Different sequences $S_{n,m_2}^{(1)}(j_1,j_2,\ldots,j_q)$ and $S_{m_2,m'}^{(2)}(j_1',j_2',\ldots,j_q')$ give different success probabilities $P_{n,m_2}^{(1)}(j_1,j_2,\ldots,j_q)$ and $P_{m_2,m'}^{(2)}(j_1',j_2',\ldots,j_q')$. We want to find the MED. The MED of two-stages search algorithm is

$$d_2(\alpha) = \min_{m_2,m',j_1,\ldots,j_q,j_1',\ldots,j_q'} \frac{d(S_{n,m_2}^{(1)}(j_1,j_2,\ldots,j_q)) + d(S_{m_2,m'}^{(2)}(j_1',j_2',\ldots,j_q'))}{P_{n,m_2}^{(1)}(j_1,j_2,\ldots,j_q)P_{m_2,m'}^{(2)}(j_1',j_2',\ldots,j_q')}$$

(26)

We optimize the total expected depth. We do not optimize the expected stage depth, because we can not verify the partial bit by neither classical nor quantum oracle. Note that $m_2$ is the length of $t_2$. We can either fix $m_2$ or optimize different choice of $m_2$. In definition $d_2(\alpha)$, see (26), we optimize the choice of $m_2$. The second stage algorithm is rescaled version of full search algorithm. Such two-stage quantum search algorithm (with depth optimizations) can be easily generalized to multi-stage quantum search algorithm.

As example, let us consider $n = 4$ two-stage search algorithm. Grover’s algorithm (one-stage search algorithm) has

success probability

$$P_4(3) = |\langle t|G_4^2|s_4\rangle|^2 \approx 0.961$$

(27)

In two-stage search algorithm, we divide the target state into two parts: $|t\rangle = |t_1\rangle|t_2\rangle$. We choose the first stage sequence as $S_{4,2}^{(1)}(1,1) = G_4G_2$. Then we measure the two qubits which do not have $D_2$ (defined in $G_2$) acting on. The probability that the measurement results reveal $|t_1\rangle$ is

$$P_{4,2}^{(1)}(4,2) \approx 0.953$$

(28)

Suppose that the measurement results are $|t_1'\rangle$ after first stage. Then we rescale the initial state as $|t_1'\rangle \otimes |s_2\rangle$. We choose the
second stage sequence as \( S_{2}^{(2)}(1, 0) = G_{2} \). Recall that the two-qubit Grover’s algorithm can find the target state in 100% probability with one Grover operator. Therefore, the second stage success probability is

\[
P_{2}^{(2)}(2, 0) = 1
\]

(29)

Then the total success probability is

\[
P_{4}^{(1)}(4, 2)P_{2}^{(2)}(2, 0) \approx 0.953
\]

(30)

The result is quiet closed to Grover’s algorithm with same number of oracles. But the depth in each stage is less then Grover’s algorithm.

Another interesting example (two-stage \( n = 4 \) search algorithm) is that sequence \( S_{4}^{(2)}(1, 2) \) gives probability 1 for finding \( t_{1} \). Combined with second stage sequence \( S_{2}^{(2)}(1, 0) \), we find a new way for \( n = 4 \) exact search algorithm [35]. We estimate \( d_{2}(\alpha) \) with \( \alpha = 1 \) for \( n = 4, 5, \ldots, 10 \) qubit search algorithm, see FIG 2. The corresponding optimal sequences are listed in Appendix B. See Appendix A for more examples (with quantum circuit diagrams) on two-stage quantum search algorithms.

IV. CRITICAL RATIOS

A. The Critical Ratio for One-stage Algorithm

Grover’s algorithm is optimal in number of query to oracle [17, 18]. Grover’s algorithm is one-stage search algorithm: no measurement within the algorithm until the end. When \( \alpha \to \infty \), we will expect \( d_{1}(\alpha) = d_{2}(\alpha) \) (no local diffusion operators). Here \( d_{1}(\alpha) \) is defined in (17). And \( d_{2}(\alpha) \) defined in (12) is the MED of Grover’s algorithm. We define the critical alpha \( \alpha_{c,1} \) for one-stage search algorithm:

\[
\alpha_{c,1} = \max \{ \alpha | d_{1}(\alpha) < d_{G}(\alpha) \}
\]

(31)

\[
S_{n,n-1}(1, 1, 1, 1) = \begin{pmatrix}
 c^{2}(c^{2} - 3s^{2}) & -cs(3c^{2} - s^{2})(c^{2} - 3s^{2}) & s(3c^{2} - s^{2}) \\
 -cs(3c^{2} - s^{2})(c^{2} - 3s^{2}) & s^{2}(2 - 3c^{2}) & c(e^{2} - 3s^{2}) \\
 s(3c^{2} - s^{2}) & c(e^{2} - 3s^{2}) & 0
\end{pmatrix}
\]

(36)

with short notations \( c = \cos \theta_{2} \) and \( s = \sin \theta_{2} \). Note that \( \sin \theta_{2} = \sqrt{2/N} \) since we choose \( m = n - 1 \). The matrix \( S_{n,n-1}(1, 1, 1) \) has the eigenvalues:

\[
\lambda_{0} = -1, \quad \lambda_{\pm} = e^{\pm i\gamma}
\]

(37)

The subscript 1 in \( \alpha_{c,1} \) means one-stage search algorithm. Below \( \alpha_{c,1} \), depth of Grover’s algorithm is not optimal. Based on the depth optimization method proposed in Sec. III B, we can give estimation on \( \alpha_{c,1} \):

**Theorem 1.** \( \alpha_{c,1} = O(n^{-1} 2^{n/2}) \).

*Proof.* The MED \( d_{1}(\alpha) \) defined in (17) is search algorithm with two different diffusion operators. One is local diffusion operator \( D_{m} \), see (7). The other is global diffusion operator \( D_{n} \), see (4). Local diffusion operator \( D_{m} \) is only acting on subspace of database. We can follow a three-dimensional subspace: target state \( |t\rangle \) defined in (9a); normalized sum of non-target state in target block \( |ntt\rangle \) defined in (9b); the normalized of rest states in database \( |u\rangle \) defined in (9c). The notations are taken from QPSA, see Sec. II B and [22, 23].

Operators \( G_{n} \) and \( G_{m} \) only change relative amplitudes of states \( |t\rangle, |ntt\rangle \) and \( |u\rangle \). Therefore, operators \( G_{n} \) and \( G_{m} \) are elements of \( O(3) \) group [31]. It is interesting to see that operator \( G_{m} \) can be viewed as a rescaled version of \( G_{n} \). In new basis \( \{|t\rangle, |ntt\rangle, |u\rangle\} \), sequence \( S_{n,m}(j) = G_{m}^{j} \) (only has local Grover operators \( G_{m} \)) has the representation

\[
S_{n,m}(j) = G_{m}^{j} = \begin{pmatrix}
 \cos(2j\theta_{2}) & \sin(2j\theta_{2}) & 0 \\
 -\sin(2j\theta_{2}) & \cos(2j\theta_{2}) & 0 \\
 0 & 0 & 1
\end{pmatrix}
\]

(32)

For example, the matrix element \( \sin(2j\theta_{2}) \) is obtained from

\[
\sin(2j\theta_{2}) = \langle t | S_{n,m}(j) | ntt \rangle
\]

(33)

The angle is defined as

\[
\sin \theta_{2} = 1/\sqrt{b}, \quad b = 2^{m}
\]

(34)

We want to estimate the critical ratio \( \alpha_{c,1} \). We consider the sequence:

\[
S_{n,n-1}(1, 1, 1) = G_{n-1}G_{n}G_{n-1}
\]

(35)

Here we choose \( m = n - 1 \). It means that the database is divided into two blocks. At basis \( \{|t\rangle, |ntt\rangle, |u\rangle\} \) defined in (9a-9c), the sequence \( S_{n,n-1}(1, 1, 1) \) has the matrix representation:

\[
\begin{pmatrix}
 cs(3c^{2} - s^{2})(c^{2} - 3s^{2}) & s(3c^{2} - s^{2}) \\
 -cs(3c^{2} - s^{2})(c^{2} - 3s^{2}) & s^{2}(2 - 3c^{2}) \\
 s(3c^{2} - s^{2}) & c(e^{2} - 3s^{2}) \\
 c^{2}(c^{2} - 3s^{2}) & 0
\end{pmatrix}
\]

with

\[
\tan \gamma = \frac{\Delta}{1 + \cos \theta_{2}}, \quad \Delta = \sqrt{3 - 2 \cos(6\theta_{2}) - \cos^{2}(6\theta_{2})}
\]

(38)

The corresponding normalized eigenvectors are denoted as
\[ |v_0\rangle \text{ (with eigenvalue } \lambda_0 \text{) and } |v_\pm\rangle \text{ (with eigenvalue } \lambda_\pm \text{).} \]

States \( |v_0\rangle \text{ and } |v_\pm\rangle \text{ have the form:} \]
\[ |v_0\rangle = \frac{1}{N_0} \left( 0, 1, \cos \theta_2 (1 - 4 \cos^2 \theta_2) \right)^T, \]
\[ |v_\pm\rangle = \frac{1}{N_\pm} \left( \mp i \sqrt{\frac{3 + \cos 6 \theta_2}{2}}, \cos 3 \theta_2, 1 \right)^T \]
\]

Notation \( T \) means transpose and \( N_0 \) and \( N_\pm \) are normalizations. Note that eigenvector \( |v_0\rangle \) (with eigenvalues \(-1\)) is orthogonal to the target state, i.e., \( \langle t | v_0 \rangle = 0 \). We can view operator \( S_{n,n-1}(1,1,1) \) as rotation combined with reflection. Rotation is around axis perpendicular to \( |t\rangle \). Rotation angle is \( \gamma \). Reflection is around plane perpendicular to \( |t\rangle \). Iteration \( S_{n,n-1}(1,1,1) \) on the initial state gives
\[ \langle t | S_{n,n-1}^j(1,1,1) | s_n \rangle = \lambda_+^j \langle t | v_+ \rangle \langle v_+ | s_n \rangle + \lambda_-^j \langle t | v_- \rangle \langle v_- | s_n \rangle \] (40)

We have \( \langle t | v_\pm \rangle = \mp i/\sqrt{2} \). Because \( N = 2^n \) is a large number, the angle \( \theta_2 \) is a small number. We can expand:
\[ \gamma = 3 \sqrt{2} \theta_2 + O(\theta_2^2), \]
\[ \langle v_\pm | s_n \rangle = \frac{1}{\sqrt{2}} + O(\theta_2) \]
\]

Substitute above relations into (40). After some algebra, we can get the success probability for finding the target state
\[ \langle t | S_{n,n-1}^j(1,1,1) | s_n \rangle \rangle^2 = \sin^2 \left( 3 \sqrt{2} \theta_2 \right) + O(\theta_2) \] (42)

Because the sandwich sequence \( S_{n,n-1}(1,1,1) \) has three oracles, we set \( j = 3j \). Then the probability difference between \( S_{n,n-1}^j(1,1,1) \) and Grover’s algorithm (with same number of oracles) is
\[ \langle t | G_1^j | s_n \rangle \rangle^2 - \langle t | S_{n,n-1}^j(1,1,1) | s_n \rangle \rangle^2 = \delta > 0 \] (43)

Here \( \delta = \alpha + \frac{1}{4} \) is a small number:
\[ \delta = O(2^{-n/2}) \] (44)

The Grover’s algorithm (with \( j \) Grover iterations) has success probability \( P_n(j) \), see (6). Then the success probability for \( S_{n,n-1}^j(1,1,1) \) sequence (with \( j = j/3 \) iterations) is \( P_n(j) \). If we want the new sequence \( S_{n,n-1}^j(1,1,1) \) to have lower expected depth than Grover’s algorithm, we can set:
\[ \frac{3(\alpha + 1) d(D_n)}{P_n(j)} > \frac{(3\alpha + 1) d(D_n) + 2 d(D_{n-1})}{P_n(j) - \delta} \] (45)

The left hand side (times \( j/3 \)) is the expect depth for Grover’s algorithm. The right hand side (times \( j = j/3 \)) is the expect depth for \( S_{n,n-1}^j(1,1,1) \) algorithm. The above inequality gives
\[ \alpha < \frac{2(d(D_n) - d(D_{n-1}))P_G}{3d(D_n)\delta} \] (46)

Diffusion operators \( D_n \) has the depth \( d(D_n) = \mathcal{O}(n) \) or \( d(D_n) = \mathcal{O}(n^2) \). Then we have
\[ \alpha_c = \mathcal{O}(n^{-1}2^{n/2}) \] (47)

This is the end of proof.

As examples, we numerically estimate \( \alpha_{c,1} \) defined in (31) for \( n = 4,5,\ldots,10 \) based on the linear depth of \( D_n \), see Appendix C and TABLE IV. Below the critical ratio \( \alpha_{c,1} \), at least two-third global diffusion operators \( D_n \) can be replaced by \( D_{n-1} \) (to have lower expected depth). The saved depth scales as \( \mathcal{O}(2^{n/2}) \).

B. The Critical Ratio for Two-stage Algorithm

Similar with one-stage search algorithm, we can define critical ratio for two-stage algorithm:
\[ \alpha_{c,2} = \max \{ \alpha | d_2(\alpha) < d_G(\alpha) \} \] (48)

Here \( d_2(\alpha) \) is MED of two-stage search algorithm, defined in (26). The two-stage search algorithm has two measurements. After first measurement, we reinitialize the state in rescaled database. Amplified amplitude of target state \( |t\rangle \) is lost in the new initialization. One can argue that
\[ d_2(\alpha) > d_1(\alpha) \] (49)

And it implies that \( \alpha_{c,2} < \alpha_{c,1} \). Analytically, we can prove:

**Theorem 2.** \( \lim_{N \to \infty} \alpha_{c,2} = 1 + \sqrt{3} \approx 2.732 \).

**Proof.** Similar with proof of Theorem 1, we construct special sequence. Then compare the expected depth of such sequence with the expected depth of Grover’s algorithm. Since we consider two-stage search algorithm, we need two sequences for two stages. Firstly, we assume that the target state \( |t\rangle \) has two parts \( |t\rangle = |t_1\rangle \otimes |t_2\rangle \), same as (22). And the length of \( t_2 \) is 2.

For first stage, we consider the sequence:
\[ S_{n,2}^j(1,1) = (G_n G_2)^j \] (50)

In first stage (by sequence \( S_{n,2}^j(1,1) \)), we find \( t_1 \) with high probability. The probability is denoted as \( P_j^{(1)} \). In second stage, we have a rescaled two-qubit search algorithm. One Grover operator \( G_2 \) can find the target state with 100\% probability. Therefore, the second stage has the sequence:
\[ S_{2}(1,0) = G_2 \] (51)

The probability of finding \( t_2 \) is \( P_j^{(2)} = 1 \).

In basis \( \{|0\rangle, |1\rangle, |u\rangle\} \) defined in (9a-9c), the sequence \( S_{n,2}(1,1) \) has the matrix representation:
\[ S_{n,2}(1,1) = \frac{1}{2} \begin{pmatrix} \cos 2\gamma & \sqrt{3} & \sin 2\gamma \\ \sqrt{3}\cos 2\gamma & -1 & \sqrt{3}\sin 2\gamma \\ -2\sin 2\gamma & 0 & 2\cos 2\gamma \end{pmatrix} \] (52)
with $\sin \gamma = 2/\sqrt{N}$. We can easily find eigenvalues and eigenvectors of $S_{n,2}^2(1,1)$. Then we can have matrix expression for $S_{n,2}^2(1,1)$. Applying $S_{n,2}^2(1,1)$ on initial state $|s_n\rangle$ (10),
\[ |(u|S_{n,2}^2(1,1)|s_n\rangle|^2 = \cos^2(\sqrt{3}j\gamma) + O(\gamma) \] (53)

Note that $|(u|S_{n,2}^2(1,1)|s_n\rangle|^2$ is the probability of finding the state in non-target block. In other words, we have
\[ P_{n,2}^{(1)} = 1 - |(u|S_{n,2}^2(1,1)|s_n\rangle|^2 \] (54)

The second stage has probability 1 (two-qubit Grover’s algorithm has probability 1). Then $P_{n,2}^{(1)}$ is also the probability of finding the target state.

The two stages designed above has total $2j + 1$ queries to oracle. In order to compare with Grover’s algorithm, we set $j = \sqrt{3}j$ ($j$ is the number of query to oracle in Grover’s algorithm). Grover’s algorithm with $j$ iterations has success probability $P_n(j)$ of finding the target state, see (6). Then the two-stage search algorithm (with sequences $S_{n,2}^2(1,1)$ and $S_2(1,0)$) can find the target state with probability $P_n(j) + \delta$. Here $\delta$ is a small number in order $\delta = O(2^{-n/2})$. If we want the two-stage search algorithm has lower expected depth than Grover’s algorithm, we need
\[ \frac{(\alpha + 1)d(D_n)}{P_n(j)} > (2\alpha + 1)d(D_n) + \frac{3}{\sqrt{3}(P_n(j) + \delta)} \] (55)

The left hand side (times $j$) is the expect depth for Grover’s algorithm (with $j$ iterations). The right hand side (times $j$) gives the expect depth for the designed two-stage search algorithm. Note that the second-stage circuit only contributes order $O(2^{-n/2})$ to the critical value $\alpha_{c,2}$, therefore we can neglect it here. Then we can solve the inequality:
\[ \alpha > 1 + \sqrt{3} - \frac{3}{d(D_n)} + O\left(2^{-n/2}\right) \] (56)

For large $N$, we have the critical ratio
\[ \lim_{N \to \infty} \alpha_{c,2} = 1 + \sqrt{3} \approx 2.732 \] (57)

End of the proof.

Theorem 2 suggests that the two-stage search algorithm can have lower expected depth than Grover’s algorithm, only when the oracle can be realized as efficiently as global diffusion operator. The real advantage for two-stage algorithm is to mitigate the error accumulations for long circuit. For examples, see FIG 2 and Appendix A and B. We numerically estimate the value $\alpha_{c,2}$ ($n = 4, 5, \ldots, 10$) based on linear scale depth of $d(D_n)$, see Appendix C and TABLE IV.

V. PARALLEL RUNNING OF QUANTUM SEARCH ALGORITHM

Now we discuss how to parallel run the quantum search algorithm on several quantum computers. The simplest idea is running low success probability search algorithm on different quantum computers. Verify the result with classical oracle and continue the algorithm until one of the quantum computer finds the target state [30]. First we can set a threshold success probability. Then we find the optimal sequence which gives the MED (the success probability is lower than the threshold success probability). We run such sequence on several quantum computers.

Another parallel running method is to combine the random guess with search algorithm, as mentioned in [23] for QPSA. For example, the target state is divided to two parts: $|t\rangle = |t_1\rangle \otimes |t_2\rangle$, same as (22). One can randomly guess the bits $t_1$. Then perform the search algorithm on bits $t_2$. Each quantum computer can pick up one guess. However, if more than half of the bit is choosing randomly, the quadratic speed up is lost. Such strategy is more efficient if some bits have higher probability (some prior information about target state).

If we want near-deterministic (the fail probability is $O(2^{-n/2})$) parallel running search algorithm, we can apply multi-stage search algorithm on different quantum computers. Suppose the target state has length $l(t) = n$. The target state is divided into $p$ parts. And each part has equally $l(t)/p$ length. Then we can assign the search algorithm on $p$ quantum computers. Each quantum computer finds one part of the target state. Combining all the results from each quantum computers, we can piece the whole solution $t$ at one time. The sequence running on each quantum computer can be found by maximizing the number of local Grover operators $G_n$ defined in (8), based on some threshold success probability $(O(1-2^{-n/2}))$. At most it requires $n$ quantum computers. Each quantum computer finds one bit of target state. However the most efficient way to find one bit of target state is by running random guess one bit search algorithm [23].

VI. CONCLUSION AND OUTLOOK

In this paper, we propose a new way to optimize the depth of quantum search algorithm. Quantum search algorithm can be realized by global and local diffusion operators. The ratio between depth of oracle and depth of global diffusion operator is important. The ratio is denoted as $\alpha$, defined in (11). The minimal practical value for $\alpha$ is 1 (in one target search algorithm). When $\alpha$ is below a threshold, we can design new algorithm (new sequence) which has lower expected depth than Grover’s algorithm. We give examples on $\alpha = 1$. In examples, our new algorithm can have around 20% lower depth than Grover’s algorithm. We also study depth optimization in multi-stage quantum search algorithm. In each stage, circuit has lower depth than Grover’s algorithm. The multi-stage quantum search algorithm gives a natural way for parallel running of the quantum search algorithm.

Ideas in this work can be easily generalized to the multi-target solution search [17]. However, the exact number of target states is required in order to find the optimal sequence. In this paper, we only consider two kinds of diffusion operators (at each stage). Further improvement is possible if more diffusion operators are working together. It will be im-
Interesting to optimize the depth of amplitude amplification algorithm [11, 12]. Grover’s algorithm is only optimal in oracle measure. New search algorithm can have lower depth than Grover’s algorithm.

ACKNOWLEDGMENTS

The authors are grateful to Professor Jin Wang and Mr. Yulun Wang. V.K. is supported by SUNY Center for Quantum Information Science at Long Island project number CSP181035.

Appendix A: Examples on $n=6$ search algorithm with depth optimizations

Different problems have different oracles. For demonstration, we can consider the simplest oracle. As mentioned in [7], oracle is single-qubit gates equivalent with $n$-qubit Toffoli gate $Λ_{n-1}(X)$. Suppose $|t⟩=|000000⟩$ ($n=6$). We can have the oracle:

$$U_t = X X X X X X$$

According to [27], $Λ_5(X)$ gate can be realized by depth 61 circuits: $d(Λ_5(X)) = 61$ (if the quantum computer can perform any single-qubit gates and any two-qubit controlled gates). In real quantum computers, the depth $d(Λ_5(X))$ may be much larger since not all qubits are connected. Nevertheless, we can set

$$d(U_t) = d(Λ_5(X)) + 2 = 63 \quad (A1)$$

The global diffusion operator ($n=6$) is also single-qubit gates equivalent to 6-qubit Toffoli gate $Λ_5(X)$ gate. We have

$$D_6 = H X X X X X$$

Therefore, we can set

$$d(D_6) = d(Λ_5(X)) + 2 = 63 \quad (A2)$$

Therefore we have the ratio $α = 1$, see (11). The local diffusion operators are acting on subspace of 6 qubits. For example, the $D_4$ diffusion operator has the quantum circuit diagram

And local diffusion operator is single-qubit gate equivalent with CNOT gate:

Accordingly, we have

$$d(D_4) = d(Λ_3(X)) + 2 = 15, \quad (A3)$$
$$d(D_2) = d(Λ_1(X)) + 2 = 3 \quad (A4)$$

Near-term quantum (or NISQ) computers are subjected to limited coherence time. We have to design low depth algorithm, or divide long circuit into shorter pieces. In $n=6$ search algorithm, Grover’s algorithm needs 6 iterations to give the maximal probability finding the target state. In experiments, we do not need to run the quantum search algorithm until the maximal probability is reached. For low depth consideration, we shall give examples on search algorithm with one or two oracles. Even in such simple scenarios, we can design better circuit by local diffusion operators.

1. One-oracle algorithm

- Grover’s algorithm. The one iteration Grover’s algorithm gives:

$$|0⟩ \xrightarrow{H} |t⟩ \xrightarrow{U_t} |s⟩ \xrightarrow{D_6}$$

Measurements at end are computational basis measurements. The whole circuit has depth

$$d(G_6) = 126 \quad (A5)$$

We can incorporate the initial Hadamard gates into $G_6$. The success probability of finding the target state is

$$P_6(1) = |⟨t|G_6|s⟩|^2 ≈ 0.1348 \quad (A6)$$
The result is better than classical algorithm. Optimal classical search has success probability 3.15%: single query followed by a random guess if the query fails (1/64 + 1/63 ≈ 3.15%). To evaluate the efficiency, we can calculate the expected depth:

$$\frac{d(G_6)}{P_6(1)} \approx 935 \quad (A7)$$

- Our optimized algorithm. In order to lower the depth, we can apply, for example, one iteration with local diffusion operator $G_4$. The one iteration local Grover operator has the circuit:

Note that $S_{6,4}(1) = G_4$ is still a 6-qubit gate, although $D_4$ is a 4-qubit gate. Notation about $S_{6,4}(1)$, see (14).

The whole circuit has depth

$$d(G_4) = 78 \quad (A8)$$

The depth is lower compared with $G_6$. The success probability finding the target state is

$$P_{6,4}(1) = |\langle t|S_{6,4}(1)|s_6\rangle|^2 \approx 0.1181 \quad (A9)$$

The success probability decreases a little bit, but still outperforms the classical case. The expected depth is:

$$\frac{d(S_{6,4}(1))}{P_{6,4}(1)} \approx 660 \quad (A10)$$

The circuit is 38% shorter than one $G_6$ iteration. The expected depth is 29% lower. Local diffusion operator may decrease the success probability, but it saves depth.

2. Two-oracle algorithm

We can apply same strategy for two-iteration search algorithm: design circuit with local diffusion operators and find the optimal one with least expected depth. We can also design two-stage quantum search algorithm. And each stage we use two oracles.

- Grover’s algorithm. Two iterations Grover’s algorithm gives:

The whole circuit has depth

$$d(G_6^2) = 252 \quad (A11)$$

The success probability of finding the target state is

$$P_6(2) = |\langle t|G_6^2|s_6\rangle|^2 \approx 0.3439 \quad (A12)$$

And the expected depth is

$$\frac{d(G_6^2)}{P_6(2)} \approx 733 \quad (A13)$$

- Our two-stage search algorithm. We divide the target state into two parts: $|t_1\rangle$ and $|t_2\rangle$. Here $t_1$ is two-bit long and $t_2$ is four-bit long. Accordingly, we can design the search algorithm which has two stages: the first stage finds $|t_1\rangle$ and the second stage finds $|t_2\rangle$. In each stage, we only have two Grover operators (local or global Grover operators).

The circuit is 38% shorter than one $G_6$ iteration. The expected depth is 29% lower. Local diffusion operator may decrease the success probability, but it saves depth.
TABLE I. Estimated MED of Grover’s algorithm, based on $\alpha = 1$. Number $\alpha$ (defined in (11)) is the ratio between oracle depth and diffusion operator depth. Diffusion operators $D_n$ have depth $d(D_n) = \{16, 32, 64, 123, 163, 203, 243\}$ with $n = 4, 5, \ldots, 10$, which comes from the decomposition of $n$-qubit Toffoli gate [27]. Single-run depth is the depth of optimal sequence (without considering the success probability). The MED $d_1(\alpha = 1)$ is defined in Eq. (12). The notation $S_n(j, 0)$ means $G^j_n$.

| $n$ | Optimal sequence | Success probability | Single-run depth | $d_1(1)$ |
|-----|------------------|---------------------|------------------|----------|
| 4   | $S_4(1, 0)$      | 0.473               | 30               | 63.47    |
| 5   | $S_5(2, 0)$      | 0.602               | 124              | 205.83   |
| 6   | $S_6(4, 0)$      | 0.816               | 504              | 617.36   |
| 7   | $S_7(6, 0)$      | 0.833               | 1464             | 1756.36  |
| 8   | $S_8(9, 0)$      | 0.861               | 2916             | 3388.03  |
| 9   | $S_9(12, 0)$     | 0.798               | 4848             | 6071.67  |
| 10  | $S_{10}(18, 0)$  | 0.838               | 8712             | 10397.28 |

TABLE II. MED of one-stage search algorithm optimized by local diffusion operators, based on $\alpha = 1$. The MED $d_1(\alpha = 1)$ is defined in Eq. (17). The depth of diffusion operator is $d(D_n) = \{8, 16, 32, 64, 123, 163, 203, 243\}$ with $n = 3, 4, \ldots, 10$. The sequence notation means $S_{n,m}(j_1, j_2, \ldots, j_q) = G^{j_1}_n G^{j_2}_m^{-1} \cdots G^{j_q}_n G^{j_1}_m$, see (14). And $j_q$ is always the number of local diffusion operator.

| $n$ | Optimal sequence | Success probability | Single-run depth | $d_1(1)$ |
|-----|------------------|---------------------|------------------|----------|
| 4   | $S_{4,3}(1, 1)$  | 0.821               | 52               | 63.32    |
| 5   | $S_{5,4}(1, 1, 1)$| 0.849               | 154              | 181.48   |
| 6   | $S_{6,4}(1, 1, 2)$| 0.755               | 360              | 476.97   |
| 7   | $S_{7,4}(1, 1, 2, 1, 2)$| 0.887              | 1173             | 1322.75  |
| 8   | $S_{8,4}(1, 1, 2, 1, 2, 1, 2)$| 0.875          | 2211             | 2527.43  |
| 9   | $S_{9,5}(1, 1, 2, 1, 2, 1, 2)$| 0.831              | 3713             | 4470.20  |
| 10  | $S_{10,5}(1, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2)$| 0.847            | 6453             | 7614.56  |

TABLE III. MED of two-stage search algorithm, based on $\alpha = 1$. The MED $d_2(\alpha = 1)$ is defined in (26). The depth of diffusion operator is $d(D_n) = \{4, 8, 16, 32, 64, 123, 163, 203, 243\}$ with $n = 2, 3, 4, \ldots, 10$.

| $n$ | Optimal sequence | Success probability | Single-run depth | $d_2(1)$ |
|-----|------------------|---------------------|------------------|----------|
| 4   | $S_{4,2}(1, 1)$  | $S_2(1, 0)$         | 0.953            | 48       | 18  | 69.25 |
| 5   | $S_{5,2}(1, 1)$  | $S_2(1, 0)$         | 0.658            | 96       | 34  | 197.51|
| 6   | $S_{6,2}(1, 1, 1)$| $S_2(1, 0)$         | 0.791            | 384      | 66  | 569.22|
| 7   | $S_{7,4}(1, 4)$  | $S_4(2, 0)$         | 0.739            | 792      | 274 | 1587.09|
| 8   | $S_{8,5}(1, 4, 1, 2)$| $S_{5,4}(1, 1, 2)$| 0.882            | 1806     | 724 | 2876.40|
| 9   | $S_{9,5}(1, 4, 1, 3, 1, 3)$| $S_{5,4}(1, 1, 2)$| 0.906            | 3542     | 884 | 4898.88|
| 10  | $S_{10,5}(1, 4, 1, 3, 1, 3, 1, 3)$| $S_{5,4}(1, 1, 2)$| 0.810            | 5485     | 1044| 8081.89|

TABLE IV. Numerical values for critical ratios $\alpha_{c,1}$ in (31) and $\alpha_{c,2}$ in (48). The results are based on the linear scale depth of diffusion operator $d(D_n)$, see [27]. Theorem 1 shows that $\alpha_{c,1}$ scales as $O(n^{-1}2^{n/2})$. Theorem 2 shows that $\alpha_{c,2}$ approaches to $1 + \sqrt{3}$ when $N = 2^n$ is very large.

| $n$ | $\alpha_{c,1}$ | $\alpha_{c,2}$ |
|-----|-----------------|-----------------|
| 4   | 2.07            | NA              |
| 5   | 4.64            | 1.21            |
| 6   | 14.65           | 1.53            |
| 7   | 29.45           | 1.76            |
| 8   | 32.88           | 2.00            |
| 9   | 45.95           | 2.17            |
| 10  | 83.97           | 2.28            |
In first stage, suppose that the two classical measurement bits are \(b_1\) and \(b_2\) \((b_1, b_2 \in \{0, 1\})\). We can not verify the partial bits \(b_1\) and \(b_2\). Since \(P_{0,4}(1, 1, 0) > 1/2\), the majority vote can be applied.

In second stage, we choose the sequence:

\[
S_4^{(2)}(2, 0) = G_4^2
\]

(A16)

And we have the circuit:

\[
\begin{array}{c}
|0\rangle \\
|0\rangle \\
|0\rangle \\
|0\rangle \\
|0\rangle \end{array}
\begin{array}{c}
X^{b_1} \\
X^{b_2} \\
H \\
H \\
H
\end{array}
\begin{array}{c}
U_t \\
U_t \\
D_4 \\
D_4 \\
G_4 \\
G_4
\end{array}
\]

The initial state is rescaled database. For example, the first stage we find \(|01\rangle\) state, then we prepare the input \(|01\rangle \otimes H^\otimes 4 |0\rangle^\otimes 4\). The probability finding \(|t_2\rangle\) is \(P_4^{(2)}(2, 0)\):

\[
P_4^{(2)}(2, 0) \approx 0.9084
\]

(A17)

The second-stage circuit has depth

\[
d(S_4^{(2)}(2, 0)) = 156
\]

(A18)

We have the expected depth

\[
d(S_6^{(1)}(1, 1, 0)) + d(S_4^{(2)}(2, 0))
\]

\[
P_{6,4}(1, 1, 0) P_4^{(2)}(2, 0)
\]

\[
\approx 707
\]

(A19)

**Appendix B: Optimal sequences based on \(\alpha = 1\)**

We present detailed numerical results plotted in FIG. 2. Suppose that we have quantum computers equipped with arbitrary single-qubit gates and arbitrary controlled two-qubit gates. It is well-known that \(n\)-qubit Toffoli gate \(\Lambda_{n-1}(X)\) can be linearly decomposed into basic operators with one ancilla qubit \([27]\). We set the depth of \(n\)-qubit Toffoli gate as \(d(\Lambda_{n-1}(X)) = \{1, 5, 13, 29, 61, 120, 160, 200, 240\}\) with \(n = 2, 3, \ldots, 10\), see \([27]\). Then the depth of diffusion operator \(D_n\) (4) is:

\[
d(D_n) = d(\Lambda_{n-1}(X)) + 2
\]

See FIG. 1. The depth of oracle \(U_t\) is characterized by the ratio \(\alpha = d(U_t)/d(D_n)\). Ratio \(\alpha\) is defined in (11). As an example, we set \(\alpha = 1\). The ratio \(\alpha = 1\) implies the simplest oracle construction, see \([7]\). We list the optimal strategy (with MED) of Grover’s algorithm \((n = 4, 5, \ldots, 10)\) in TABLE I. When \(N = 2^n\) is large, the optimized iteration number in Grover’s algorithm converges to \([0.583\sqrt{N}]\). And the success probability converges to 0.844. The results are independent with \(\alpha\), see \(d_k(\alpha)\) (13).

We numerically find the optimal sequence (optimised by local diffusion operator). Similarly, we set \(\alpha = 1\). The MED is given by \(d_1(\alpha = 1)\), see (17). The results are listed in TABLE II. We also numerically find the optimal sequence for two-stage search algorithm. The MED is given by \(d_k(\alpha = 1)\), see (26). The results are listed in TABLE III. In general, different values \(\alpha\) will give different optimal sequences. It is clearly that both the single-run depth (depth of the optimal sequence) and expected depth in TABLE II and III are smaller than the Grover’s algorithm (TABLE I). In practice, once \(\alpha\) is known, one can guess the optimal sequence based on results with small \(n\). For example, when \(n\) is large, the sequence is closed to (assume that \(n\) is even)

\[
S_{n,n/2}(1, 1, 2, \ldots, 1, 2, 1, 2) = G_{n/2}G_nG_{n/2}^2 \cdots G_nG_{n/2}^2
\]

(B2)

See TABLE II. The repetition number of \(G_nG_{n/2}^2\) can be found either by numerical or analytical ways.

**Appendix C: Examples on Critical Ratios**

The ratio \(\alpha\) defined in (11) is important parameter. If \(\alpha \to \infty\), Grover’s algorithm is optimal in depth. Critical ratios \(\alpha_{c,1}\) in (31) and \(\alpha_{c,2}\) in (48) are threshold values. Below \(\alpha_{c,1}\) (or \(\alpha_{c,2}\), we can find lower expected depth than Grover’s algorithm. Diffusion operator \(d(D_n)\) is single-qubit gate equivalent with \(n\)-qubit Toffoli gate \(\Lambda_{n-1}(X)\) gate. We can set \(d(\Lambda_{n-1}(X)) = \{1, 5, 13, 29, 61, 120, 160, 200, 240\}\) with \(n = 2, 3, \ldots, 10\), see \([27]\). Based on the depth optimization method defined in Sec. III B and III C, we find the critical ratios \(\alpha_{c,1}\) and \(\alpha_{c,2}\) in TABLE IV.

---

[1] M. A. Nielsen and I. L. Chuang, Quantum computation and quantum information (2010).

[2] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, SIAM journal on Computing 26, 1510 (1997).
[3] L. K. Grover, Physical Review Letters 79, 325 (1997).
[4] P. R. Giri and V. E. Korepin, Quantum Information Processing 16, 315 (2017).
[5] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, B. Campbell, et al., Nature 508, 500 (2014).
[6] C. Ballance, T. Harty, N. Linke, M. Sepiol, and D. Lucas, Physical Review Letters 117, 060504 (2016).
[7] C. Figgatt, D. Maslov, K. Landsman, N. M. Linke, S. Debnath, and C. Monroe, Nature Communications 8, 1918 (2017).
[8] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, et al., Nature 574, 505 (2019).
[9] J. Preskill, Quantum 2, 79 (2018).
[10] A. W. Cross, L. S. Bishop, S. Sheldon, P. D. Nation, and J. M. Gambetta, Physical Review A 100, 032328 (2019).
[11] L. K. Grover, Physical Review Letters 80, 4329 (1998).
[12] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, Contemporary Mathematics 305, 53 (2002).
[13] A. Tulsi, Physical Review A 86, 042331 (2012).
[14] A. Tulsi, Physical Review A 91, 052307 (2015).
[15] P. Kim, D. Han, and K. C. Jeong, Quantum Information Processing 17, 339 (2018).
[16] S. Jaques, M. Naehrig, M. Roetteler, and F. Virdia, arXiv preprint arXiv:1910.01700 (2019).
[17] M. Boyer, G. Brassard, P. Hoyer, and A. Tapp, Fortschritte der Physik: Progress of Physics 46, 493 (1998).
[18] C. Zalka, Physical Review A 60, 2746 (1999).
[19] G. Kato, Physical Review A 72, 032319 (2005).
[20] Z. Jiang, E. G. Rieffel, and Z. Wang, Physical Review A 95, 062317 (2017).
[21] L. K. Grover and J. Radhakrishnan, in Proceedings of the seventeenth annual ACM symposium on Parallelism in algorithms and architectures (ACM, 2005) pp. 186–194.
[22] V. E. Korepin and L. K. Grover, Quantum Information Processing 5, 5 (2006).
[23] V. E. Korepin, Journal of Physics A: Mathematical and General 38, L731 (2005).
[24] V. E. Korepin and J. Liao, Quantum Information Processing 5, 209 (2006).
[25] M. E. Morales, T. Tlyachev, and J. Biamonte, Physical Review A 98, 062333 (2018).
[26] T. J. Yoder, G. H. Low, and I. L. Chuang, Physical Review Letters 113, 210501 (2014).
[27] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, Physical Review A 52, 3457 (1995).
[28] Y. He, M.-X. Luo, E. Zhang, H.-K. Wang, and X.-F. Wang, International Journal of Theoretical Physics 56, 2350 (2017).
[29] P. Gokhale, J. M. Baker, C. Duckering, N. C. Brown, K. R. Brown, and F. T. Chong, arXiv preprint arXiv:1905.10481 (2019).
[30] R. M. Gingrich, C. P. Williams, and N. J. Cerf, Physical Review A 61, 052313 (2000).
[31] V. E. Korepin and B. C. Vallilo, Progress of theoretical physics 116, 783 (2006).
[32] B.-S. Choi and V. E. Korepin, Quantum Information Processing 6, 243 (2007).
[33] K. Zhang and V. Korepin, Quantum Information Processing 17, 143 (2018).
[34] V. E. Korepin and Y. Xu, International Journal of Modern Physics B 21, 5187 (2007).
[35] Z. Diao, Physical Review A 82, 044301 (2010).