Quantum Speedup for Inferring the Value of Each Bit of a Solution State in Unsorted Databases Using a Bio-Molecular Algorithm on IBM Quantum's Computers

Weng-Long Chang, Wen-Yu Chung, Chun-Yuan Hsiao, Renata Wong, Ju-Chin Chen, Mang Feng, and Athanasios V. Vasilakos

***Abstract*—In this paper, we propose a bio-molecular algorithm with O(n^2) biological operations, O(2^{n-1}) DNA strands, O(n) tubes and the longest DNA strand, O(n), for inferring the value of a bit from the only output satisfying any given condition in an unsorted database with 2^n items of n bits. We show that the value of each bit of the outcome is determined by executing our bio-molecular algorithm n times. Then, we show how to view a bio-molecular solution space with 2^{n-1} DNA strands as an eigenvector and how to find the corresponding unitary operator and eigenvalues for inferring the value of a bit in the output. We also show that using an extension of the quantum phase estimation and quantum counting algorithms computes its unitary operator and eigenvalues from bio-molecular solution space with 2^{n-1} DNA strands. Next, we demonstrate that the value of each bit of the output solution can be determined by executing the proposed extended quantum algorithms n times. To verify our theorem, we find the maximum-sized clique to a graph with two vertices and one edge and the solution b that satisfies $b^2 \equiv 1$ (mod 15) and $1 < b <$ (15/2) using IBM Quantum's backend.**

***Index Terms*—Data structures and algorithms, molecular algorithms, quantum algorithms, NP-complete problems.**

Manuscript received August 25, 2021; revised October 27, 2021 and November 14, 2021; accepted November 22, 2021. Date of publication November 25, 2021; date of current version April 1, 2022. This work was supported by the National Science Foundation of the Republic of China under Grant Ministry of Science and Technology (MOST) 105-2221-E-151-040. The work of Mang Feng was supported by the Key Research and Development Project of Guangdong Province under Grant 2020B0303300001. *(Corresponding author: Renata Wong.)*

Weng-Long Chang, Wen-Yu Chung, Chun-Yuan Hsiao, and Ju-Chin Chen are with the Department of Computer Science and Information Engineering, National Kaohsiung University of Science and Technology, Kaohsiung 80778, Taiwan (e-mail: changwl@cc.kuas.edu.tw; wychung@nkust.edu.tw; cyhsiao@nkust.edu.tw; jc.chen@nkust.edu.tw).

Renata Wong is with the Physics Division, National Center for Theoretical Sciences, Taipei 10617, Taiwan (e-mail: renata.wong@protonmail.com).

Mang Feng is with the Wuhan Institute of Physics and Mathematics, Chinese Academy of Sciences, Wuhan 430071, China, and also with the Research Center for Quantum Precision Measurement, Guangzhou Institute of Industry Technology, Guangzhou 511458, China (e-mail: mangfeng@wipm.ac.cn).

Athanasios V. Vasilakos is with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350116, China (e-mail: th.vasilakos@gmail.com).

This article has supplementary downloadable material available at https://doi.org/10.1109/TNB.2021.3130811, provided by the authors.

Digital Object Identifier 10.1109/TNB.2021.3130811

I. Molecular Algorithms for Finding the Single Solution Among 2^n Unsorted Items

A CLAUSE is a formula that is of the form $x_1\ \theta_1 x_2 \ldots x_{n-1}\theta_{n-1}\ x_n$, where each bit x_d for $1 \le d \le n$ is a Boolean variable or its negation and $\theta_k \in \{\vee, \wedge, \bar{\vee}, \bar{\wedge}\}$ for $1 \le k \le n-1$. Elements in $\{\vee, \wedge, \bar{\vee}, \bar{\wedge}\}$ are subsequently the logic **OR**, **AND**, **NOR** and **NAND** operations. A search problem from [1]–[6] is to that from $\{x_1 x_2 \ldots x_{n-1} x_n | \forall\ x_d \in \{0, 1\}$ for $1 \le d \le n\}$ just an item satisfies any given condition and we would like to find the only item (*answer*). A common formulation of the search problem is to that any given *oracular* function (any given condition) is

$$O_f(x_1, x_2, \ldots, x_{n-1}, x_n) = C_1\beta_1 C_2 \ldots C_{m-1}\beta_{m-1} C_m, \quad (1)$$

where each C_j for $1 \le j \le m$ is a clause or its negation and $\beta_y \in \{\vee, \wedge, \bar{\vee}, \bar{\wedge}\}$ for $1 \le y \le m-1$. Its domain is $\{x_1 x_2 \ldots x_{n-1} x_n | \forall x_d \in \{0, 1\}$ for $1 \le d \le n\}$ and its range is $\{0, 1\}$. The search problem is to find a unique answer λ of n bits from its domain that satisfies the condition $O_f(\lambda) = 1$, whereas for all other inputs of n bits from the same domain, ω, for $0 \le \omega \le 2^n - 1$ and $\omega \ne \lambda$, $O_f(\omega) = 0$.

A. Binary Search Trees for Representing Problem's Domain

We use a binary search tree in Fig. 1 to represent the *structure* of the domain. In the tree, a node stands for a bit of one element in the domain. The root of the tree is x_1. The value of the *left* branch of each node indicates that the value of the corresponding bit is 0 while the value of the *right* branch of each node indicates that the value of the corresponding bit is 1. The binary search tree in Fig. 1 contains 2^n subtrees and each subtree encodes one element in the domain. For example, the first subtree (x_1)--^{0}--(x_2)--^{0}-- $\ldots(x_{n-1})$--^{0}--(x_n)--^{0}-- encodes the first element $x_1^0\ x_2^0\ \ldots\ x_{n-1}^0\ x_n^0$. The second subtree (x_1)--^{0}--(x_2)--^{0}-- $\ldots(x_{n-1})$--^{0}--(x_n)--^{1}-- encodes the second

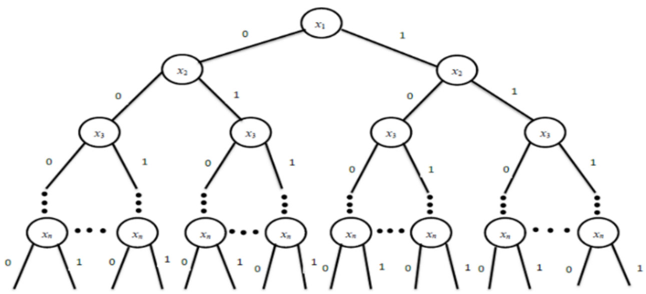


Fig. 1. A binary search tree encodes the domain of any given oracular function $O_f(x_1, x_2, \ldots, x_{n-1}, x_n)$ in (1).

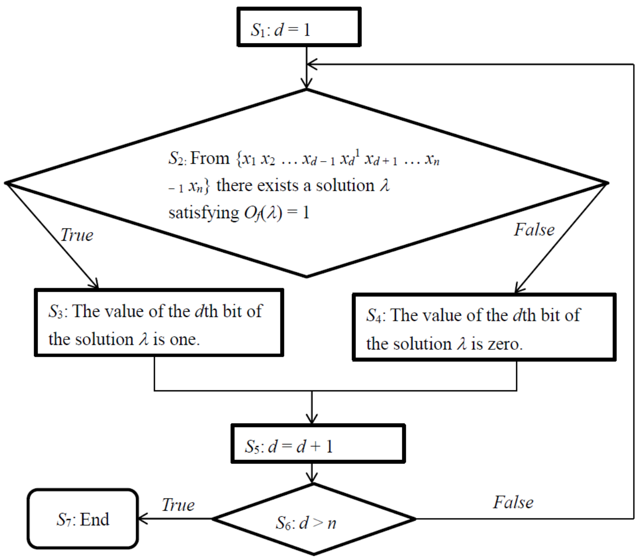


Fig. 2. Logical flowchart for finding the output state λ of length n bits and satisfying $O_f(\lambda) = 1$.

element $x_1^0\ x_2^0\ \ldots\ x_{n-1}^0\ x_n^1$. The last subtree (x_1)--^{1}--(x_2)--^{1}-- $\ldots(x_{n-1})$--^{1}--(x_n)--^{1}-- encodes the last element $x_1^1\ x_2^1\ \ldots\ x_{n-1}^1 x_n^1$.

B. Architecture for Computing the Value of Each Bit of the Output State in the Problem

Fig. 2 shows the architecture for finding the value of each bit of the solution λ with n bits that satisfies $O_f(\lambda) = 1$. The first execution of the *first* statement is to set the initial value of the loop index variable d to one. Each execution of the *second* statement is to judge whether there exists a solution in $\{x_1\ x_2 \ldots x_{d-1}\ x_d^1 x_{d+1} \ldots x_{n-1} x_n |\ \forall x_k \in \{0, 1\}$ for $1 \le k \le n$ and $k \ne d\}$. If there exists a solution λ satisfying $O_f(\lambda) = 1$, then each execution of the *third* statement will indicate that the value of the dth bit of λ is 1. Otherwise, each execution of the *fourth* statement will indicate that the value of the dth bit of λ is 0. Next, each execution of the *fifth* statement increases the loop index variable d, while each execution of the *sixth* statement checks the condition of the loop and each execution of the *seventh* statement terminates the logical flowchart.

C. Introduction to Bio-Molecular Operations

We will use the following bio-molecular operations cited in [7]–[9] to construct molecular solutions for inferring the only answer in the search problem in (1). For implementations of the operations see Supplemental Material.

Definition 1: Given set $X = \{x_n x_{n-1} \ldots x_2 x_1 |\ \forall x_d \in \{0, 1\}$ for $1 \le d \le n\}$ and a bit x_j, the bio-molecular operation, "Append-Tail", appends x_j onto the end of every element in set X. The formal representation is written as Append-Tail $(X, x_j) = \{x_n x_{n-1} \ldots x_2 x_1\ x_j | \forall x_d \in \{0, 1\}$ for $1 \le d \le n$ and $x_j \in \{0, 1\}\}$.

Definition 2: Given set $X = \{x_n x_{n-1} \ldots x_2 x_1 |\ \forall x_d \in \{0, 1\}$ for $1 \le d \le n\}$, the bio-molecular operation "Discard(X)" resets X to an empty set and can be represented as "$X = \emptyset$".

Definition 3: Given set $X = \{x_n x_{n-1} \ldots x_2 x_1 |\ \forall x_d \in \{0, 1\}$ for $1 \le d \le n\}$, the bio-molecular operation "Amplify(X, $\{X_i\}$)" creates a number of identical copies X_i of set X, and then discards X with the help of "Discard(X)".

Definition 4: Given set $X = \{x_n x_{n-1}\ \ldots\ x_2 x_1 |\ \forall x_d \in \{0, 1\}$ for $1 \le d \le n\}$ and a bit x_j, the bio-molecular *extract* operation has two kinds of representation. The first representation is $+(X, x_j^1) = \{x_n x_{n1}\ \ldots\ x_j^1\ \ldots\ x_2\ x_1 | \forall x_d \in \{0, 1\}$ for $1 \le d \ne j \le n\}$ and $-(X, x_j^1) = \{x_n\ x_{n-1}\ \ldots\ x_j^0\ \ldots x_2\ x_1 | \forall x_d \in \{0, 1\}$ for $1 \le d \ne j \le n\}$ if the value of x_j is equal to one. The second representation is $+(X, x_j^0) = \{x_n x_{n-1}\ \ldots\ x_j^0\ \ldots\ x_2\ x_1 | \forall x_d \in \{0, 1\}$ for $1 \le d \ne j \le n\}$and $-(X, x_j^0) = \{x_n\ x_{n-1}\ \ldots\ x_j^1\ \ldots\ x_2\ x_1 | \forall x_d \in \{0, 1\}$ for $1 \le d \ne j \le n\}$ if the value of x_j is equal to zero.

Definition 5: Given m sets $X_1\ \ldots\ X_m$, the bio-molecular *merge* operation is $\cup(X_1,\ \ldots,\ X_m) = X_1 \cup \ldots \cup X_m$.

Definition 6: Given set $X = \{x_n x_{n-1}\ \ldots\ x_2 x_1 |\ \forall x_d \in \{0, 1\}$ for $1 \le d \le n\}$, the bio-molecular operation "Detect(X)" returns *true* if X is not an empty tube. Otherwise, it returns *false*.

D. Molecular Algorithms for Inferring the Value of Each Bit of the Solution Among 2^n Unsorted Items

From (1), two subsets are $\{x_1 x_2 \ldots x_{d-1}\ x_d^1 x_{d+1} \ldots x_{n-1} x_n |\ \forall x_k \in \{0, 1\}$ for $1 \le k \le n$ and $k \ne d\}$ and $\{x_1 x_2\ \ldots\ x_{d-1} x_d^0\ x_{d+1}\ \ldots\ x_{n-1} x_n | \forall\ x_k \in \{0, 1\}$ for $1 \le k \le n$ and $k \ne d\}$ and their union constitutes the domain. If the only solution of the search problem in (1) lies in the first subset, then its dth bit is 1. Otherwise, its dth bit is 0. We propose the following molecular algorithms to infer the value of each bit of the single solution in the search problem in (1). The first parameter is an empty tube (an empty set) Y_0 that is regarded as the input tube (set). The second parameter n represents the number of bits in its domain and the third parameter m represents the number of clauses (conditions). The fourth parameter θ_k $(1 \le k \le n-1)$ represents logic operations of two operands in each clause. The fifth parameter β_y $(1 \le y \le m-1)$ represents logic operations of two clauses. Each tube in the **procedure Infer-the-value-of-each-bit** is an empty tube that is regarded as an auxiliary storage.

Procedure Infer-the-value-of-each-bit(Y_0, n, m, θ_k, β_y)
(1) **For** $d = 1$ **to** n
  (1a) **Bio-Molecular-Solution**(Y_0, n, d).
  (1b) **Condition**(Y_0, n, m, θ_k, β_y, d).
  (1c) **If** (detect(Y_0)) **then**
    (1d) The value of the dth bit of the solution is 1.

(1e) **Else**
(1f) The value of the dth bit of the solution is 0.
(1g) Discard(Y_0).
End For
EndProcedure

The procedure **Infer-the-value-of-each-bit** makes use of the following subroutines:

Procedure Bio-Molecular-Solution(Y_0, n, d)
(1) **For** $k = 1$ **to** n
(1a) Amplify(Y_0, Y_1, Y_2).
(1b) **If** (the value of k is not equal to the value of d) **Then**
(1c) Append-Tail(Y_1, x_k^1). (1d) Append-Tail(Y_2, x_k^0).
(1e) **Else**
(1f) Append-Tail(Y_1, x_k^1). (1g) Discard(Y_2).
EndIf
(1h) $Y_0 = \cup(Y_1, Y_2)$.
End For
EndProcedure

Procedure Condition(Y_0, n, m, θ_k, β_y, d)
(1) **For** $j = 1$ **to** m
(2) **For** $k = 1$ **to** $n - 1$
(2a) **If** (the value of k is equal to 1) **Then**
(2b) **Gate**(Y_0, x_1, x_2, $r_{j,k}$, θ_k, d).
(2c) **Else**
(2d) **Gate**(Y_0, $r_{j,k-1}$, x_{k+1}, $r_{j,k}$, θ_k, d).
EndIf
End For
End For
(3) **For** $y = 1$ **to** $m - 1$
(3a) **If** (the value of y is equal to 1) **Then**
(3b) **Gate**(Y_0, $r_{1,n-1}$, $r_{2,n-1}$, s_y, β_y, d).
(3c) **Else**
(3d) **Gate**(Y_0, s_{y-1}, $r_{y+1,n-1}$, s_y, β_y, d).
EndIf
EndFor
(4) $T_5 = +(Y_0, s_{m-1}^1)$ and $T_6 = -(Y_0, s_{m-1}^1)$.
(5) $Y_0 = \cup(Y_0, T_5)$.
(6) Discard(T_6).
EndProcedure

Procedure Gate(Y_0, p_1, p_2, re, op, d).
(1) $T_1 = +(Y_0, p_1^1)$ and $T_0 = -(Y_0, p_1^1)$.
(2) $T_{1,1} = +(T_1, p_2^1)$ and $T_{1,0} = -(T_1, p_2^1)$.
(3) $T_{0,1} = +(T_0, p_2^1)$ and $T_{0,0} = -(T_0, p_2^1)$.
(4) **If**(op is a logic **OR** operation) **Then**
(4a) **OR**(Y_0, $T_{1,1}$, $T_{1,0}$, $T_{0,1}$, $T_{0,0}$, p_1, p_2, re).
(5) **Else If**(op is a logic **AND** operation) **Then**
(5a) **AND**(Y_0, $T_{1,1}$, $T_{1,0}$, $T_{0,1}$, $T_{0,0}$, p_1, p_2, re).
(6) **Else If**(op is a logic **NOR** operation) **Then**
(6a) **NOR**(Y_0, $T_{1,1}$, $T_{1,0}$, $T_{0,1}$, $T_{0,0}$, p_1, p_2, re).
(7) **Else If**(op is a logic **NAND** operation) **Then**
(7a) **NAND**(Y_0, $T_{1,1}$, $T_{1,0}$, $T_{0,1}$, $T_{0,0}$, p_1, p_2, re).
EndProcedure

The logic **OR**, **AND**, **NOR** and **NAND** operations can be implemented using the **procedure Gate**(Y_0, p_1, p_2, re, op, d). Details on the respective implementations are given in the Supplemental Material.

Lemma 1: Logical flowchart for finding the value of each bit in the solution of n bits in length and satisfying $O_f(x_1, x_2, \ldots, x_{n-1}, x_n) = 1$ in Fig. 2 is implemented using the **procedure Infer-the-value-of-each-bit**(Y_0, n, m, θ_k, β_y), where $O_f(x_1, x_2, \ldots, x_{n-1}, x_n)$ is any given condition of a search problem in (1-1) and O_f: $\{x_1x_2 \ldots x_{n-1}x_n | \forall x_d \in \{0, 1\}$ for $1 \le d \le n\} \to \{0, 1\}$.

Proof: Step (1) is the main loop and the value of its index variable d is from 1 through n. On the pth execution of Step (1) for $2 \le p \le n$, the value of d is equal to p. Next, on the pth execution of Step (1a), the procedure **Bio-Molecular-Solution** (Y_0, n, d) is called to construct $\{x_1x_2 \ldots x_{p-1}x_p^1\ x_{p+1} \ldots x_{n-1}x_n | \forall\ x_k \in \{0, 1\}$ for $k \ne p$ and $1 \le k \le n\}$ that is encoded by 2^{n-1} DNA strands in tube Y_0. On the pth execution of Step (1b), the procedure **Condition** (Y_0, n, m, θ_k, β_y, d) is called to implement $O_f(x_1x_2 \ldots x_{p-1}x_p^1\ x_{p+1} \ldots x_{n-1}x_n)$ so that its evaluated result with $O_f(x_1x_2 \ldots x_{p-1}x_p^1\ x_{p+1} \ldots x_{n-1}x_n) = 1$ is stored in tube Y_0 and other evaluated results with $O_f(x_1x_2 \ldots x_{p-1}x_p^1\ x_{p+1} \ldots x_{n-1}x_n) = 0$ are discarded. Next, on the pth execution of Step (1c), if a *true* is returned from the Detect operation, then the value of the pth bit of the solution is 1 from the pth execution of Step (1d). Otherwise, the value of the pth bit of the solution is 0 from the pth execution of Step (1f). Next, on the pth execution of Step (1g), the Discard operation is used to reset Y_0 to an empty tube. From the statements above, it is at once inferred that the logical flowchart for finding the value of each bit in the solution with the n bits satisfying $O_f(x_1, x_2, \ldots, x_{n-1}, x_n) = 1$ in Fig. 2 is implemented by using the **procedure Infer-the-value-of-each-bit** (Y_0, n, m, θ_k, β_y). ■

II. Extension of Phase Estimation and Quantum Counting to Infer the Value of Each Bit of a Solution

The proposed molecular algorithm **Infer-the-value-of-each-bit** (Y_0, n, m, θ_k, β_y) infers the value of each bit of the output state that satisfies an arbitrary *oracular* function $O_f(x_1, x_2, \ldots, x_{n-1}, x_n)$ with m conditions. In this section we demonstrate how to, step by step, implement the proposed molecular algorithm **Infer-the-value-of-each-bit** (Y_0, n, m, θ_k, β_y) for labelling solution(s) and non-solution(s) using quantum circuits, and how to view bio-molecular solution space with 2^{n-1} DNA strands as an eigenvector. We also describe how an extension of the quantum phase estimation algorithm and the quantum counting algorithm finds the corresponding unitary operator and eigenvalues. Finally, we give a complexity assessment of the extension of phase estimation and quantum counting routines.

A. Quantum Circuits for Implementing Molecular Circuits for Labelling Solutions Among 2^{n-1} Subtrees

We use a unique *computational basis vector* with 2^n-tuples of binary numbers to represent each element in $\{x_1x_2 \ldots x_{n-1}x_n | \forall\ x_d \in \{0, 1\},\ 1 \le d \le n\}$. The corresponding vector for the *first* element $x_1^0\ x_2^0 \ldots x_{n-1}^0 x_n^0$ is $(\begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}_{1\times 2^n}^T)$ and so on with the corresponding vector for the *last* element $x_1^1\ x_2^1 \ldots x_{n-1}^1 x_n^1$ being $(\begin{bmatrix} 0 & 0 & \cdots & 1 \end{bmatrix}_{1\times 2^n}^T)$.

We assume that the initial state vector ($|\phi_0\rangle$) is $(\otimes_{k=1}^{d-1}\left|x_k^0\right\rangle)\otimes(\left|x_d^1\right\rangle)\otimes(\otimes_{k=d+1}^{n}\left|x_k^0\right\rangle)$. After applying a unitary operator $(\otimes_{k=1}^{d-1}H)\otimes(I_{2\times 2})\otimes(\otimes_{k=d+1}^{n}H)$, in which H is a two-dimensional Hadamard gate and $I_{2\times 2}$ is a two-dimensional identify operator, to the initial state vector ($|\phi_0\rangle$), the following new state vector is obtained

$$|\phi_1\rangle = \frac{1}{\sqrt{2^{n-1}}}(\otimes_{k=1}^{d-1}(\left|x_k^0\right\rangle + \left|x_k^1\right\rangle))\otimes(\left|x_d^1\right\rangle)\otimes(\otimes_{k=d+1}^{n}(\left|x_k^0\right\rangle+\left|x_k^1\right\rangle)) \quad (2)$$

$|\phi_1\rangle$ encodes $\{x_1x_2 \ldots x_{d-1}x_d^1x_{d+1} \ldots x_{n-1}\, x_n|\forall x_k \in \{0, 1\}$ for $1 \le k \le n$ and $k \ne d\}$ and the amplitude of each subtree is $(\frac{1}{\sqrt{2}^{n-1}})$. Based on the molecular algorithm **Bio-Molecular-Solution** (Y_0, n, d), since 2^{n-1} DNA strands in tube Y_0 encode $\{x_1x_2 \ldots x_{d-1}x_d^1\, x_{d+1} \ldots x_{n-1}x_n|\forall\, x_k \in \{0, 1\}$ for $1 \le k \le n$ and $k \ne d\}$ that correspond to the 2^{n-1} subtrees in the binary search tree in Fig. 1-1, they are encoded by $|\phi_1\rangle$.

Based on **Lemma 1** through **Lemma 9** (see Supplemental Material for details), the molecular algorithm **Infer-the-value-of-each-bit** uses biological operations to implement each condition of any given oracular function $O_f(x_1, x_2, \ldots, x_{n-1}, x_n)$. We use auxiliary quantum bits $|r_{j,k}>$ ($1 \le j \le m$, $1 \le k \le n-1$) to store the outcome of executing the kth logic operation in the jth clause. We also apply auxiliary quantum bits $|s_y>$ ($1 \le y \le m-1$) to store the outcome of executing the yth logic operation between the yth clause and the ($y+1$)th clause. We assume that an auxiliary state vector $|a>$ is $(\otimes_{y=m-1}^{1}|s_y>)\otimes(\otimes_{j=m}^{1}\otimes_{k=n-1}^{1}\ |r_{j,k}>)$.

Since the state vector $|\phi_1\rangle$ encodes $\{x_1x_2 \ldots x_{d-1}x_d^1\, x_{d+1} \ldots x_{n-1}x_n|\forall\, x_k \in \{0, 1\}, 1 \le k \le n, k \ne d\}$, items that are solutions in $|\phi_1\rangle$ are referred to as *marked* states and those that are not solutions are referred to as *unmarked* states. A special gate (quantum circuit), the so-called Oracle, is applied to label the marked and unmarked states. The Oracle O multiplies the probability amplitude of the solution(s) by -1 and leaves any other amplitude unchanged:

$$O : |\phi_1\rangle\, |a\rangle \to (-1)^{O_f(x_1\cdots x_{d-1}x_d^1x_{d+1}\ldots x_n)}\, |\phi_1\rangle\, |a\rangle\,, \quad (3)$$

If $O_f(x_1x_2\ \ldots\ x_{d-1}x_d^1\ x_{d+1}\ \ldots\ x_{n-1}x_n) = 1$, then the probability amplitude of the solution(s) is multiplied by -1. The probability amplitude of the non-solution(s) is multiplied by 1. As the quantum circuit of the Oracle is made up of *CCNOT gates*, *NOT gates* and a *CNOT* gate, the Oracle is a *unitary* operator.

Let $N = 2^{n-1}$ and let S be the number of solution(s) in ($|\phi_1\rangle$). We create two superpositions comprising uniformly distributed computational basis states that are made up of a set of marked states and another set of unmarked states in ($|\phi_1\rangle$), respectively

$$|\varphi\rangle = \frac{1}{\sqrt{N-S}} \sum\nolimits_{O_f(x_1\cdots x_{d-1}x_d^1x_{d+1}\ldots x_n)=0} \left|x_1\cdots x_{d-1}x_d^1x_{d+1}\ldots x_n\right\rangle ., \quad (4)$$

$$|\lambda\rangle = \frac{1}{\sqrt{S}} \sum\nolimits_{O_f(x_1\cdots x_{d-1}x_d^1x_{d+1}\ldots x_n)=1} \left|x_1\cdots x_{d-1}x_d^1x_{d+1}\ldots x_n\right\rangle . \quad (5)$$

Since the inner product of $|\varphi\rangle$ and $|\lambda\rangle$ is 0 and $|\varphi\rangle$ and $|\lambda\rangle$ are unit vectors, $|\varphi\rangle$ and $|\lambda\rangle$ form an orthonormal basis of a two-dimensional Hilbert space. With $|\varphi\rangle$ and $|\lambda\rangle$, we rewrite (2)

$$|\phi_1\rangle = (\frac{\sqrt{N-S}}{\sqrt{N}}\,|\varphi\rangle + \frac{\sqrt{S}}{\sqrt{N}}\,|\lambda\rangle) \quad (6)$$

This is to say that in an orthonormal basis of a two-dimensional Hilbert space that is formed by ($|\varphi\rangle$) and ($|\lambda\rangle$), the initial state ($|\phi_1\rangle$) in (6) is represented as a two-dimensional unit vector. The state vector $|\phi_2\rangle = O\,|\phi_1\rangle$ obtained by applying the Oracle to (6) can then be expressed as $|\phi_2\rangle = (\frac{\sqrt{N-S}}{\sqrt{N}}\,|\varphi\rangle + (-\frac{\sqrt{S}}{\sqrt{N}}\,|\lambda\rangle))$. The angle between $|\phi_2\rangle$ and $|\varphi\rangle$ is equal to $\theta/2$. The Oracle O is hence equivalent to a reflection about axis ($|\varphi\rangle$).

An operator U that amplifies the probability amplitudes of marked states while decreasing any other amplitudes is given as

$$U = 2\,|\phi_1\rangle\,\langle\phi_1| - I_{2^n\times 2^n}. \quad (7)$$

(2) indicates that the state vector ($|\phi_1\rangle$) $= ((\otimes_{k=1}^{d-1}H)\otimes(I_{2\times 2})\otimes\ (\otimes_{k=d+1}^{n}H))\ \ ((\otimes_{k=1}^{d-1}\left|x_k^0\right\rangle)\otimes\ (\left|x_d^1\right\rangle)\otimes(\otimes_{k=d+1}^{n}\left|x_k^0\right\rangle))$. For convenience of presentation, we assume that $Q = ((\otimes_{k=1}^{d-1}H)\otimes(I_{2\times 2})\otimes(\otimes_{k=d+1}^{n}H))$ and $(|\phi_0\rangle) = ((\otimes_{k=1}^{d-1}\left|x_k^0\right\rangle)\otimes(\left|x_d^1\right\rangle)\otimes(\otimes_{k=d+1}^{n}\left|x_k^0\right\rangle))$. Therefore, (7) can be further refined to

$$U = 2Q\,|\phi_0\rangle\,\langle\phi_0|\,Q - Q I_{2^n\times 2^n} Q = Q(2\,|\phi_0\rangle\langle\phi_0| - I_{2^n\times 2^n})Q. \quad (8)$$

Equation (8) provides the key for the realization of the operator U. We require two Q gates and a *controlled phase shift* gate P. The transformational rule of P is quite simple: it leaves all the probability amplitudes unchanged except that of $((\otimes_{k=1}^{d-1}\left|x_k^0\right\rangle)\otimes(\left|x_d^1\right\rangle)\otimes(\otimes_{k=d+1}^{n}\left|x_k^0\right\rangle))$ whose sign is inverted. Because Q and P are both unitary operators, U is also unitary.

B. Determining the Eigenvalue and the Corresponding Eigenvector of the Extension of the Grover Operator in a Two-Dimensional Hilbert Space Spanned by Sets of Marked and Unmarked States Among 2^{n-1} Subtrees Encoded by 2^{n-1} DNA Strands

Given an extension of the Grover operator as $G = (U)(O)$, we have that $|\phi_3\rangle = (U)(O)\ (|\phi_1\rangle) = (U)\ (|\phi_2\rangle) = (G)(|\phi_1\rangle)$, i.e.

$$|\phi_3\rangle = (\frac{\sqrt{N-S}}{\sqrt{N}} \times (\frac{N - 4\times S}{N})\,|\varphi\rangle + \frac{\sqrt{S}}{\sqrt{N}} \times (\frac{3\times N - 4\times S}{N})\,|\lambda\rangle) \quad (9)$$

The angle between $|\phi_3\rangle$ and $|\phi_1\rangle$ is θ. Basic trigonometry provides for the projection of $|\phi_1>$ onto the axes

$$\cos(\theta/2) = \frac{\sqrt{N-S}}{\sqrt{N}}/1 = \frac{\sqrt{N-S}}{\sqrt{N}},$$
$$\sin(\theta/2) = \frac{\sqrt{S}}{\sqrt{N}}/1 = \frac{\sqrt{S}}{\sqrt{N}} \quad (10)$$

$|\phi_1\rangle$ and $|\phi_3\rangle$ can be expressed as $|\phi_1\rangle = (\cos(\theta/2), \sin(\theta/2))$ and $|\phi_3\rangle = (G|\phi_1\rangle) = (\cos(\theta + (\theta/2)), \sin(\theta + (\theta/2)))$. With this we obtain the following unitary operator, which is an extension of the Grover operator in the basis $\{|\varphi\rangle, |\lambda\rangle\}$

$$G = (\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}_{2\times 2}). \tag{11}$$

This indicates that G acts on an orthonormal basis of a two-dimensional Hilbert space that is formed by $(|\varphi\rangle)$ and $(|\lambda\rangle)$.

We assume that a (2×2) matrix G^+ is the conjugate transpose of G. From G's characteristic equation it follows that the two solutions (eigenvalues) are $\cos(\theta) + \sqrt{-1}\times \sin(\theta) = e^{\sqrt{-1}\times\theta}$ and $\cos(\theta) + (-\sqrt{-1}\times \sin(\theta)) = e^{-\sqrt{-1}\times\theta}$. We assume that the corresponding eigenvector of G in the basis $\{|\varphi\rangle, |\lambda\rangle\}$ is $|V\rangle = (\begin{bmatrix} p \\ q \end{bmatrix}_{2\times 1})$. Therefore, $G|V\rangle = e^{\pm\sqrt{-1}\times\theta}|V\rangle$. From this we have that $p\times \cos(\theta) - q\times \sin(\theta) = p\times (\cos(\theta) \pm\sqrt{-1}\times \sin(\theta))$ and $p\times \sin(\theta) + q\times \cos(\theta) = q\times (\cos(\theta) \pm \sqrt{-1}\times \sin(\theta))$. This leads to the two equations with the solutions $|V_1\rangle = (\frac{e^{\sqrt{-1}\times\gamma}}{\sqrt{2}}\begin{bmatrix} \sqrt{-1} \\ 1 \end{bmatrix}_{2\times 1})$ and $|V_2\rangle = (\frac{e^{\sqrt{-1}\times\gamma}}{\sqrt{2}}\begin{bmatrix} -\sqrt{-1} \\ 1 \end{bmatrix}_{2\times 1})$, where γ is a real number. Therefore, both $|V_1\rangle$ and $|V_2\rangle$ are the corresponding eigenvectors of G in the basis $\{|\varphi\rangle, |\lambda\rangle\}$

C. Determining the Value of a Bit of the Only Solution Among 2^{n-1} Subtrees Encoded by 2^{n-1} DNA Strands

The extension of quantum counting and phase estimation is used to decide the number of solutions for $\{x_1x_2 \ldots x_{d-1}x_d^1 x_{d+1} \ldots x_{n-1}x_n|\forall\, x_k \in \{0, 1\}, 1 \le k \le n, k \ne d\}$. In Fig. 3, the number of the quantum bits for the upper quantum register is $t = n + p$ where p is the number of auxiliary quantum bits so that the probability of measuring the solution is at least $1 - (\frac{1}{2(2^p - 2)})$ [3]–[6]. The circuit in Fig. 3 aims to find phase θ, which is equivalent to finding the phase ratio $p_r \in [0, 1)$: $\theta = 2 \times \pi \times p_r$. If p_r is an integer multiple of $(1/2^t)$ that is equivalent to a binary expansion of t bits, then the state of the first register is read out with probability 1 [3]–[6].

In Fig. 3, if an eigenvalue generated from controlled Grover operations is $e^{\sqrt{-1}\times\theta}$, then we use controlled Grover operations followed by the IQFT to find the best approximation of t bits to the real number θ. Otherwise, we use the QFT instead of IQFT. Since both $\sin(\theta/2)$ and N are *known*, by determining θ we can also find $S = N\times (\sin(\theta/2))^2$. After Grover's iteration, the quantum state of the second register is not changed and is still $(|\phi_1\rangle) = \frac{1}{\sqrt{2}^{n-1}}(\otimes_{k=1}^{d-1}(|x_k^0\rangle + |x_k^1\rangle)) \otimes (|x_d^1\rangle)\otimes(\otimes_{k=d+1}^{n}(|x_k^0\rangle + |x_k^1\rangle))$.

D. Complexity Assessment for the Extension of Phase Estimation and Quantum Counting

Lemma 2: In determining the number of solutions in $\{x_1x_2 \ldots x_{d-1}x_d^1x_{d+1} \ldots x_{n-1}\, x_n|\forall x_k \in \{0, 1\}, 1 \le k \le n, k \ne d\}$, the space complexity of the extension of phase estimation and quantum counting with the probability of measuring the solution being at least $1 - (\frac{1}{2(2^p-2)})$ is $O(t + n) = O(n + p + n)$ quantum bits, where $t = n + q$ and q is the number of auxiliary quantum bits such that the probability of measuring the solution is at least $1 - (\frac{1}{2(2^p-2)})$.

Proof: See Supplemental Material.

Lemma 3: For computing the number of solutions in $\{x_1\, x_2 \ldots x_{d-1}\, x_d^1x_{d+1} \ldots x_{n-1}x_n|\ \forall x_k \in \{0, 1\}, 1 \le k \le n, k \ne d\}$, the time complexity of the *ideal* case in the extension of phase estimation and quantum counting with the probability of measuring the solution being 1 is $O(t^2 + 2t + (n-1)) = O((n+p)^2 + 2(n+p) + (n-1))$ quantum gates, where $t = n + p$ and p is the number of auxiliary quantum bits such that the probability of measuring the solution is at least $1 - (\frac{1}{2(2^p-2)})$.

Proof: See Supplemental Material

Lemma 4: In determining the number of solutions in $\{x_1x_2 \ldots x_{d-1}x_d^1x_{d+1} \ldots x_{n-1}\, x_n|\forall x_k \in \{0, 1\}, 1 \le k \le n, k \ne d\}$, for the extension of quantum phase estimation and quantum counting with the probability of measuring the solution being $(\frac{1}{2^{2t}}\frac{\sin^2(\frac{2\pi(2^t\theta - i)}{2})}{\sin^2(\frac{2\pi(\theta - \frac{i}{2^t})}{2})})$, the time complexity of the *practical* case is $O(t^2 + 2t + (n-1)) = O((n+p)^2 + 2(n+p) + (n-1))$ quantum gates, where $t = n + p$ and p is the number of auxiliary quantum bits such that the probability of measuring the solution is at least $1 - (\frac{1}{2(2^p-2)})$.

Proof: See Supplemental Material

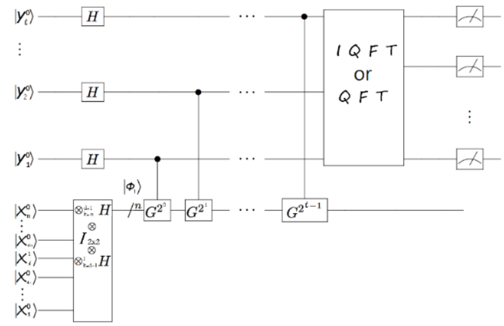


Fig. 3. Quantum circuit for finding the phase θ. The controlled G operators are sequentially applied to the initial state of the lower register $(|\phi_1\rangle) = (\frac{\sqrt{N-S}}{\sqrt{N}}|\varphi\rangle + \frac{\sqrt{S}}{\sqrt{N}}|\lambda\rangle)$.

III. Experimentally Solving the Clique Problem in a Graph With Two Vertices and One Edge on IBM Quantum's Simulator

Mathematically, a *clique* for a graph G^1 with n vertices and m edges is a *complete* sub-graph of G^1 [10]. The clique problem [10] is to find a *maximum-sized* clique in G^1 that is a *maximum-sized* subset V^1 of vertices with size r. In Fig. 4, the left-most picture is a graph with two vertices and one edge, which is the simplest case of the clique problem. The graph consists of two vertices $\{v_1, v_2\}$ and one edge $\{(v_1, v_2)\}$. The set of possible solutions consist of 2^2 elements. Every element corresponds to a subset of vertices (a possible clique).

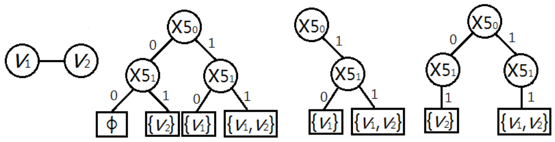


Fig. 4. Architecture for solving the clique problem (simplest case).

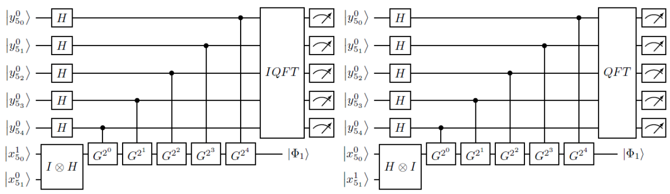


Fig. 5. Deciding whether the first vertex (left)/the second vertex (right) lies in the maximum-sized clique.

We assume that $\beta = \{x5_0\ x5_1 | \forall x5_d \in \{0, 1\}, 0 \le d \le 1\}$ is a set of 2^2 possible choices (elements). For the sake of presentation, we further assume that $x5_d^0$ denotes that $x5_d = 0$ and $x5_d^1$ denotes that $x5_d = 1$. If an element $x5_0x5_1$ in β is a legal clique and $x5_d = 1$ for $0 \le d \le 1$, then $x5_d^1$ indicates that the $(d+1)$th vertex is within the legal clique. If an element $x5_0x5_1$ in β is a legal clique and $x5_d = 0$ for $0 \le d \le 1$, then $x5_d^0$ indicates that the $(d+1)$th vertex is not within the legal clique.

The second picture from the left in Fig. 4 shows a binary search tree representing $\{x5_0x5_1 | x5_k \in \{0, 1\}, 1 \le k \le 2\}$. The subtree $(x5_0)$--^{0}--$(x5_1)$--^{0}-- represents $x5_0^0\ x5_1^0$ that encodes the empty set clique. The subtree $(x5_0)$--^{0}--$(x5_1)$--^{1}-- represents $x5_0^0\ x5_1^1$ that encodes the clique $\{v_2\}$. The subtree $(x5_0)$--^{1}--$(x5_1)$--^{0}-- represents $x5_0^1\ x5_1^0$ that encodes the clique $\{v_1\}$. The subtree $(x5_0)$--^{1}--$(x5_1)$--^{1}-- represents $x5_0^1\ x5_1^1$ encoding the clique $\{v_1, v_2\}$. The third picture on the left is the binary search tree half that represents $\{x5_0^1\ x5_1 | x5_1 \in \{0, 1\}\}$ and encodes two possible solutions $\{v_1\}$ and $\{v_1, v_2\}$. The fourth picture is the other half of the binary search tree that represents $\{x5_0\ x5_1^1 | x5_0 \in \{0, 1\}\}$ and encodes two possible solutions $\{v_2\}$ and $\{v_1, v_2\}$.

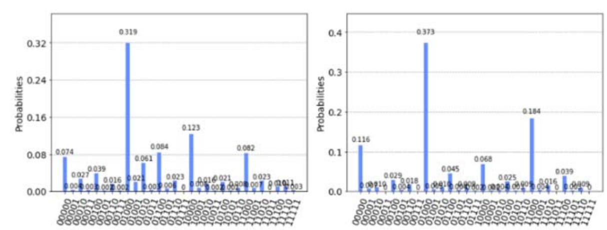
We use the quantum circuits in Fig. 5 to decide whether the first and the second vertex, respectively, lie in the maximum clique. In both circuits, the first (upper) register contains five quantum bits, initially in the state $|0>$. In Fig. 5 (left) the two quantum bits of the second (lower) register are initially in the state $|x5_0^1\rangle \otimes |x5_1^0\rangle$. In Fig. 5 (right) the two quantum bits of the second register are initially in the state $|x5_0^0\rangle \otimes |x5_1^1\rangle$. In Fig. 5 (left) we apply the unitary operator $(I \otimes H)$ to the second register. This indicates that it encodes the two subtrees $(x5_0)$--^{1}--$(x5_1)$--^{0}-- and $(x5_0)$--^{1}--$(x5_1)$--^{1}--. In Fig. 5 (right) we apply the unitary operator $(H \otimes I)$ to the second register. This implies that it encodes the other two subtrees $(x5_0)$--^{0}--$(x5_1)$--^{1}-- and $(x5_0)$--^{1}--$(x5_1)$--^{1}--.

Next, we apply the unitary operator $(H \otimes H \otimes H \otimes H \otimes H)$ to the first register $(|y5_0^0> \otimes |y5_1^0> \otimes |y5_2^0> \otimes |y5_3^0> \otimes |y5_4^0>)$ in Fig. 5 (left), followed by controlled G operations with G raised to successive powers of two to the second register. We use $e^{\sqrt{-1}\times\theta}$ with real θ as the eigenvalue of G in Fig. 5 (left), and $e^{-\sqrt{-1}\times\theta}$ with real θ as another eigenvalue of G in Fig. 5 (right). Next, we apply the *inverse* quantum Fourier transform (IQFT) to the first register in Fig. 5 (left), and the quantum Fourier transform (QFT) to the first register in Fig. 5 (right). Finally, we measure the first register in both figures in the computational basis.

Fig. 6. Measurement results for Fig. 5 (left) and (right), respectively.

Fig. 7. Deciding whether the first vertex lies in the maximum-sized clique on the backend *Manila*.

Fig. 8. Inferring whether the second vertex lies in the maximum-sized clique on the backend *Manila*.



(a) (b)

Fig. 9. Measurement result for (a) Fig. 5 (left) and (b) Fig. 5 (right) on the Manila backend.

We use the open quantum assembly language version 2.0 on IBM's simulator to implement the quantum circuits in Fig. 5. Both programs are included in the Supplemental Material. Fig. 6 shows the two experimental results. The two measured states are 01000 with probability 1. Since phase ratios for the phase θ in Fig. 5 are equal to $8/2^5, \theta = \times 2\pi 8/2^5 = \pi/2$ in both cases. Since $\sin(\theta/2) = \sin(\pi/4) = 1/\sqrt{2}$ in both figures, the number of solution(s) S is equal to $2(1/\sqrt{2})^2$ in both. This indicates that there exists one solution in the two subtrees $(x5_0)$--^{1}--$(x5_1)$--^{0}-- and $(x5_0)$--^{1}--$(x5_1)$--^{1}-- and one solution in the other two subtrees $(x5_0)$--^{0}--$(x5_1)$--^{1}-- and $(x5_0)$--^{1}--$(x5_1)$--^{1}--. This is to say that both the first vertex and the

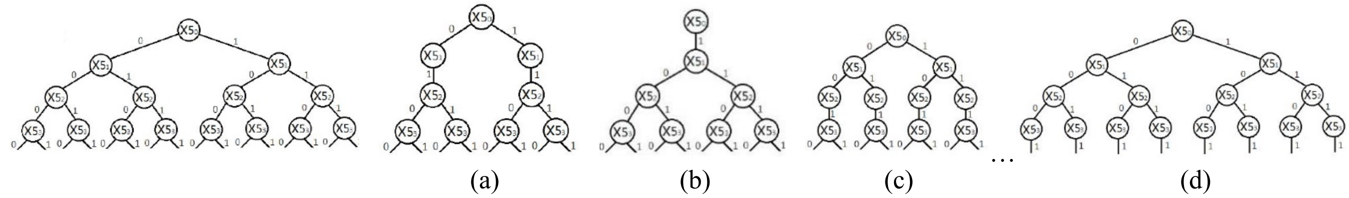


Fig. 10. The architecture for solving the example $b^2 \equiv 1 \pmod{15}$. (a): Inferring the value of the second bit in the integer solution satisfying $b^2 \equiv 1 \pmod{15}$. (b): Deciding the value of the first bit in the integer solution satisfying $b^2 \equiv 1 \pmod{15}$. (c): Determining the value of the third bit in the integer solution satisfying $b^2 \equiv 1 \pmod{15}$. (d): Computing the value of the fourth bit in the integer solution satisfying $b^2 \equiv 1 \pmod{15}$.

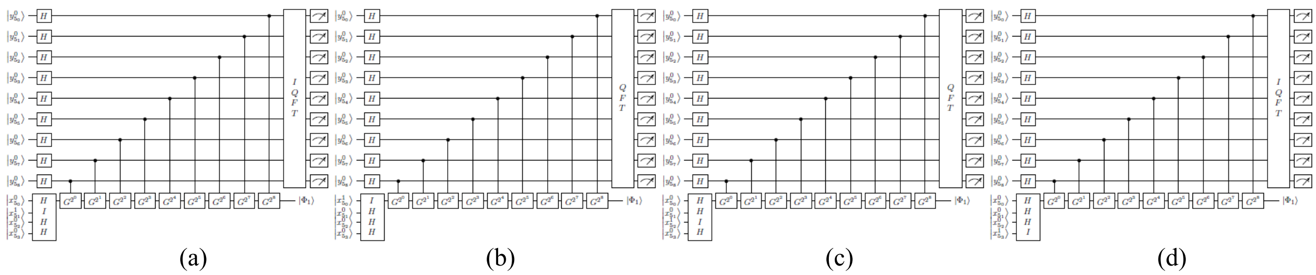


Fig. 11. Quantum circuits for inferring the value of the second (a), the first (b), the third (c), and the fourth (d) bit in the integer solution satisfying $b^2 \equiv 1 \pmod{15}$.

second vertex lie in the maximum-sized clique. Therefore, the maximum-sized clique is $\{v_1, v_2\}$.

IV. Experimentally Solving the Clique Problem in a Graph With Two Vertices and One Edge on the Backend Manila

The backend *Manila* of **IBM** is a new quantum processor with five quantum bits. We wrote the third program and the fourth program to execute subsequently Fig. 5 on the backend *Manila*. Fig. 7 shows the quantum circuit implementation of Fig. 5 (left), while Fig. 8 shows the quantum circuit implementation of Fig. 5 (right). In Fig. 7 and Fig. 8, the $P(\lambda)$ gate is a phase gate with a real parameter λ. The corresponding matrix of this gate is $\begin{pmatrix} 1 & 0 \\ 0 & e^{\sqrt{-1}\times\lambda} \end{pmatrix}$. The gate leaves $|0>$ unchanged and modifies $|1>$ to $(e^{\sqrt{-1}\times\lambda})\ |1>$.

See Supplemental Material for the programs for Fig. 5 (left) and 7, and for Fig. 5 (right) and 8, respectively. In Fig. 9(a) and Fig. 9(b), respectively, the state of the first register of Fig. 5 are read out. The two states are 01000 with probability 0.319 and 0.373, respectively. Since the phase ratios of the phase θ in Fig. 7 and 8 are equal to $8/2^5$, $\theta = (\times 2\pi\, 8/2^5) = \pi/2$ in both figures.

As both figures, $\sin(\theta/2)$=$\sin(\pi/4) = 1/\sqrt{2}$, therefore, the number of solution(s) $S = 2 \times (1/\sqrt{2})^2$in both Fig. 7 and 8. Hence, there exists one solution in the two subtrees $(x5_0)$--^{1}--$(x5_1)$--^{0}-- and $(x5_0)$--^{1}--$(x5_1)$--^{1}-- and one solution in the other two subtrees $(x5_0)$--^{0}--$(x5_1)$--^{1}-- and $(x5_0)$--^{1}--$(x5_1)$--^{1}--. Therefore, both the first and the second vertex lie in the maximum-sized clique $\{v_1, v_2\}$.

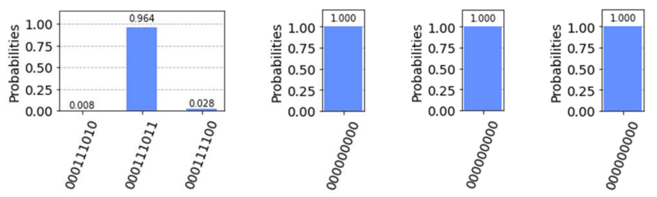
V. Experimentally Finding Solution B Satisfying $B^2 \equiv 1$ (Mod 15) and $1 < B < (15/2)$ on QASM Simulator

Let a natural number P be of length n bits and let a function $O_f(x_1, x_2, \ldots, x_{n-1}, x_n)$ be defined as $\{x_1 x_2 \ldots x_{n-1} x_n |$ $0 \le x_1\ x_2 \ldots x_{n-1}\ x_n \le P\} \rightarrow \{(x_1 x_2 \ldots x_{n-1} x_n)^2 \pmod P\}$. Four integer solutions that satisfy $O_f(x_1, x_2, \ldots, x_{n-1}, x_n) = (x_1 x_2 \ldots x_{n-1} x_n)^2 \equiv 1 \pmod P$ are, respectively, b, $P - b$, 1 and $P-$ 1, where $1 < b < (P/2)$ and $(P/2) < P - b < P - 1$. This is a special case of quadratic congruence (mod P) and it is still a NP-complete problem [10]. Solving this problem is equivalent to finding the only integer solution $x_1 x_2 \ldots x_{n-1} x_n$ for $1 < x_1 x_2 \ldots x_{n-1} x_n < (P/2)$ that satisfies $O_f(x_1, x_2, \ldots, x_{n-1}, x_n) = (x_1 x_2 \ldots x_{n-1} x_n)^2 \equiv 1 \pmod P$. Consider the example where in $\{x5_0 x5_1\ x5_2 x5_3 | \forall x5_d \in \{0, 1\}$ for $0 \le d \le 3\}$, only the element $x5_0^0 x5_1^1\ x5_2^0 x5_3^0$ satisfies $4^2 \equiv 1 \pmod{15}$ and $1 < 4 < (15/2)$. The example is a simple case of quadratic congruence (mod P), where $P = 15$.

Fig. 10 shows a binary search tree encoding all possible input states. In Fig. 10(a), $\{x5_0\ x5_1^1 x5_2 x5_3 |\ \forall x5_d \in \{0, 1\}$ for $0 \le d \le 3$ and $d \ne 1\}$ is encoded, while in Fig. 10(b), $\{x5_0^1 x5_1 x5_2\ x5_3 | \forall x5_d \in \{0, 1\}$ for $1 \le d \le 3\}$. Fig. 10(c) encodes $\{x5_0 x5_1 x5_2^1\ x5_3 | \forall x5_d \in \{0, 1\}$ for $0 \le d \le 3$ and $d \ne 2\}$, and Fig. 10(d) encodes $\{x5_0 x5_1 x5_2\ x5_3^1 | \forall x5_d \in \{0, 1\}$ for $0 \le d \le 2\}$.

The four quantum circuits in Fig. 11 compute the values of the *second*, the *first*, the *third* and the *fourth* bit of the solution state. In all the figures, the first register consists of 9 quantum bits, initially in the state $|0>$, and the 4 quantum bits of the second register are initially $\left|x5_0^0\right\rangle \otimes \left|x5_1^1\right\rangle \otimes \left|x5_2^0\right\rangle \otimes \left|x5_3^0\right\rangle$, $\left|x5_0^1\right\rangle \otimes \left|x5_1^0\right\rangle \otimes \left|x5_2^0\right\rangle \otimes \left|x5_3^0\right\rangle$, $\left|x5_0^0\right\rangle \otimes \left|x5_1^0\right\rangle \otimes \left|x5_2^1\right\rangle \otimes \left|x5_3^0\right\rangle$ and $\left|x5_0^0\right\rangle \otimes \left|x5_1^0\right\rangle \otimes \left|x5_2^0\right\rangle \otimes \left|x5_3^1\right\rangle$, respectively. For encoding each subtree from Fig. 10(a) through Fig. 10(d), we respectively use the unitary operators $(H \otimes I \otimes H \otimes H)$, $(I \otimes H \otimes H \otimes H)$, $(H \otimes H \otimes I \otimes H)$ and $(H \otimes H \otimes H \otimes I)$ on the second register in Fig. 11.

Next, the unitary operator $H^{\otimes 9}$ is applied to the first register in Fig. 11. After that follow the controlled G operations with G raised to successive powers of two applied on the second register. The respective eigenvalues of G are $e^{\sqrt{-1}\times\theta}$



(a) (b) (c) (d)

Fig. 12. The measurement outcome for the circuit in (a) Fig. 11(a), (b) Fig. 11(b), (c) Fig. 11(c), (d) Fig. 11(d).

in Fig. 11(a), $e^{-\sqrt{-1}\times\theta}$ in Fig. 11(b), $e^{-\sqrt{-1}\times\theta}$ in Fig. 11(c) and $e^{\sqrt{-1}\times\theta}$ in Fig. 11(d), with θ a real number. Next, the IQFT and the QFT are applied respectively on nine quantum bits on the first register in Fig. 11. Finally, the first register is measured in the computational basis.

We wrote four programs (see Supplemental Material) to implement the four quantum circuits in Fig. 11 on the IBM simulator. The respective measurement results appear in Fig. 12(a) through Fig. 12(d).

In Fig. 12(a), the state 000111011 is measured with probability 0.964. In Fig. 12(b) through 12(d), the measured outcome is 000000000 with probability 1. Thus, for computing the value of the *second* bit of the solution, the *approximate value* of the phase rate is (5^9/ 2^9) and the *approximate value* of its phase θ is $\times 2\pi\, 5^9/2^9$. Since $\sin(\theta/2) = \sin(\times\pi\, 5^9/2^9) = 0.3541635252$, therefore also $0.3541635252 = \sqrt{S}/\sqrt{8}$. With this we obtain $S = 1.00345442$. As S is the number of solutions in $\{x5_0 x5_1^1 x5_2\ x5_3 | \forall x5_d \in \{0, 1\}, 0 \le d \le 3, d \ne 1\}$, the approximate value for S is equal to 1. This indicates that the number of solutions is 1 and the value of the second bit $x5_1$ in the solution is 1.

Similarly, for computing the value of the *first*, *third* and *fourth* bit of the solution, the approximate value of the phase rate is ($0/2^9$) and the approximate value of θ is ($2\pi \times 0/2^9$) = 0. Since $\sin(\theta/2) = \sin(0) = 0 = \sqrt{S}/\sqrt{8}$, the approximate value of S is equal to zero. This implies that there is no solution in $\{x5_0^1 x5_1\ x5_2 x5_3 | \forall x5_d \in \{0, 1\}, 1 \le d \le 3\}$, $\{x5_0\ x5_1 x5_2^1 x5_3 |\ \forall x5_d \in \{0, 1\}, 0 \le d \le 3, d \ne 2\}$ and $\{x5_0 x5_1\ x5_2 x5_3^1 | \forall\ x5_d \in \{0, 1\}, 0 \le d \le 2\}$. Thus, the solution is $x5_0^0\ x5_1^1 x5_2^0\ x5_3^0$.

VI. Conclusion

Based on **Lemma 1S** through **Lemma 9S** (see Supplemental Material for details), we can infer the value of each bit in the solution state of a search problem using $O(n^2+n^2\times m+m\times n)$ biological operations, $O(2^{n-1})$ DNA strands, O(1) tubes and the longest DNA strand, $O(n+n\times m)$. From Fig. 3 and from **Lemma 2** through **Lemma 4** we have that the approximation of the phase θ can be determined with a very high probability of success. Since N is known and θ can be determined, we can find the number of solution(s) $S = (N\times\ (\sin(\theta/2))^2)$. Therefore, if the number of solution(s) is not equal to zero, then the answer lies in $\{x_1 x_2\ \ldots\ x_{d-1} x_d^1\ x_{d+1}\ \ldots\ x_{n-1} x_n | \forall x_k \in \{0, 1\}, 1 \le k \le n, k \ne d\}$ and the dth bit of the solution must be a 1. Otherwise, the dth bit of the solution must be a 0. This indicates that for solving the same problem the molecular algorithm **Infer-the-value-of-each-bit** can be implemented by using an extension of the quantum phase estimation algorithm and the quantum counting algorithm. An interesting open question is whether the proposed bio-quantum algorithm can be extended to solve a problem with r solutions, where $r > 2$.

References

[1] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Phys. Rev. Lett.*, vol. 79, no. 2, pp. 325–328, Jul. 1997.

[2] C. Zalka, "Grover's quantum searching algorithm is optimal," *Phys. Rev. A, Gen. Phys.*, vol. 60, no. 4, p. 2746, 1999.

[3] W.-L. Chang and A. V. Vasilakos, *Fundamentals of Quantum Programming in IBM's Quantum Computers*. Germany: Springer, Jan. 2021.

[4] G. Brassard, P. Hoyer, and A. Tapp, "Quantum counting," in *Proc. 25th Int. Colloq. Automata, Lang., Program.*, Jul. 1998, pp. 820–831.

[5] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[6] S. Imre and F. Balazs, *Quantum Computation and Communications: An Engineering Approach*. Hoboken, NJ, USA: Wiley, 2005.

[7] L. M. Adleman, "Molecular computation of solutions to combinatorial problems," *Science*, vol. 266, no. 5187, pp. 1021–1024, 1994.

[8] W.-L. Chang and A. V. Vasilakos, *Molecular Computing: Towards a Novel Computing Architecture for Complex Problem Solving*. Germany: Springer, Jun. 2014.

[9] M. Amos, *Theoretical and Experimental DNA Computation*. Germany: Springer, Apr. 2006.

[10] M. R. Garey and D. S. Johnson, *Computer and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman, 1979.