Quantum Algorithms and Mathematical Formulations of Biomolecular Solutions of the Vertex Cover Problem in the Finite-Dimensional Hilbert Space

Weng-Long Chang*, Ting-Ting Ren, and Mang Feng

***Abstract*—In this paper, it is shown that the proposed quantum algorithm for implementing Boolean circuits generated from the DNA-based algorithm solving the vertex-cover problem of any graph G with m edges and n vertices is the optimal quantum algorithm. Next, it is also demonstrated that mathematical solutions of the same biomolecular solutions are represented in terms of a unit vector in the finite-dimensional Hilbert space. Furthermore, for testing our theory, a nuclear magnetic resonance (NMR) experiment of three quantum bits to solve the simplest vertex-cover problem is completed.**

***Index Terms*—Data structure and algorithm, quantum algorithms, molecular algorithms, nuclear magnetic resonance.**

I. Introduction

IN 1961 AND 1982 Feynman [1], [2] respectively gave the possible chance of a molecular computer and a quantum computer that perhaps are faster than the standard Turing machines [3]. In 1994 Adleman [4] succeeded in solving an instance of the Hamiltonian path problem just by handling DNA strands. In 1985 Deutsch [5] denoted a general model of quantum computation. An interesting open question is to ask what the mathematical solutions of biomolecular solutions for dealing with any NP-Complete problem are. Our motivation is to find the answer of the interesting open question.

Our major contributions in this journal paper are as follows.

- The proposed quantum algorithm for implementing Boolean circuits generated from the DNA-based algorithm solving the vertex-cover problem of any graph G with m edges and n vertices is the optimal quantum algorithm.
- It is demonstrated that mathematical solutions of the same biomolecular solutions are represented in terms of a unit vector in the finite-dimensional Hilbert space.
- It is proved that biological operations with DNA strands and quantum gates with quantum bits can each other simulate for solving the same problem.
- A nuclear magnetic resonance (NMR) experiment of three quantum bits to solve the simplest vertex-cover problem is completed.

II. The Formal Model of Computation

In this section, the vertex cover problem of any graph with m edges and n vertices in [6] and biological operations in [7] are introduced. Next, quantum bits and quantum gates in [8] are illustrated, and they will be used to design quantum circuits to show that biomolecular solutions for solving it are represented in terms of a unit vector in the finite-dimensional Hilbert space.

A. Definition of the Vertex Cover Problem

It is supposed that G is a graph and $G = (V, E)$, where V is a set of n vertices in G and E is a set of m edges in G. Also it is assumed that V is $\{v_1, \ldots, v_n\}$ and E is $\{(v_a, v_b) | v_a$ and v_b are, respectively, elements in $V\}$. Mathematically, a *vertex cover* of graph G is a subset $V^1 \subseteq V$ of vertices such that for each edge (v_a, v_b) in E, at least one of v_a and v_b belongs to V^1 [6]. **Definition 2-1** cited in [6] is used to denote the vertex-cover problem of graph G.

Definition 2-1: The vertex cover problem of graph G with n vertices and m edges means finding a minimum-sized vertex cover in G.

Consider a graph G^1 to contain three vertices $\{v_3, v_2, v_1\}$ and two edges $\{(v_2, v_1), (v_3, v_1)\}$. All of the vertex covers in G^1 are $\{v_1\}, \{v_2, v_1\}, \{v_3, v_1\}, \{v_3, v_2\}$, and $\{v_3, v_2, v_1\}$. The minimum-sized vertex cover for G^1 is $\{v_1\}$.

B. Introduction of Biological Molecular Operations

DNA (deoxyribonucleic acid) includes polymer chains which are commonly regarded as DNA strands in [7]. Each strand may be made of a sequence of nucleotides, or bases, attached to a sugar-phosphate "backbone." The four DNA nucleotides are adenine, guanine, cytosine, and thymine, commonly abbreviated to A, G, C, and T, respectively. Double-stranded DNA may be denatured into single strands by heating the solution to a temperature determined by the composition of the strand in [7]. Annealing is the reverse of melting, whereby a solution of single strands is cooled, and allowing complementary strands to bind together in [7]. From a biological standpoint, all sequences generated to represent bits must be checked to ensure that the DNA strands that they encode do not form unwanted secondary

Manuscript received June 03, 2014; revised October 22, 2014; accepted November 03, 2014. Date of publication December 05, 2014; date of current version February 27, 2015. This work was supported by National Science Foundation of Republic of Chinaunder Grants NSC-102-2221-E-151-031- and NSC-102-2622-E-151-013-CC3. *Asterisk indicates corresponding author.*

*W.-L. Chang is with the Department of Computer Science and Information Engineering, National Kaohsiung University of Applied Sciences, Kaohsiung 807, Taiwan (e-mail: changwl@cc.kuas.edu.tw).

T.-T. Ren is with Wuhan Institute of Physics and Mathematics, Chinese Academy of Sciences, Wuhan, 430071, China (e-mail: ttren@wipm.ac.cn).

M. Feng is with Wuhan Institute of Physics and Mathematics, Chinese Academy of Sciences, Wuhan, 430071, China (e-mail: mangfeng@wipm.ac.cn).

Digital Object Identifier 10.1109/TNB.2014.2375356

structures with one another. The following biomolecular operations cited in [7] will be applied to construct molecular solutions for the vertex cover problem of any graph with m edges and n vertices. Their implementation can be found in [7].

Definition 2-2: Given set $X = \{x_n x_{n-1} \ldots x_2 x_1 | \forall x_d \in \{0,1\}$ for $1 \leq d \leq n\}$ and a bit x_j, the biomolecular operation "Append-Head" appends x_j onto the head of every element in set X. The formal representation is written as Append-Head$(X, x_j) = \{x_j x_n x_{n-1} \ldots x_2 x_1 | \forall x_d \in \{0,1\}$ for $1 \leq d \leq n$ and $x_j \in \{0,1\}\}$.

Definition 2-3: Given set $X = \{x_n x_{n-1} \ldots x_2 x_1 | \forall x_d \in \{0,1\}$ for $1 \leq d \leq n\}$ and a bit x_j, the biomolecular 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 $1 \leq d \leq n$ and $x_j \in \{0,1\}\}$.

Definition 2-4: Given set $X = \{x_n x_{n-1} \ldots x_2 x_1 | \forall x_d \in \{0,1\}$ for $1 \leq d \leq n\}$, the biomolecular operation "Discard(X)" sets X to be an empty set and can be represented as "$X = \emptyset$."

Definition 2-5: Given set $X = \{x_n x_{n-1} \ldots x_2 x_1 | \forall x_d \in \{0,1\}$ for $1 \leq d \leq n\}$, the biomolecular operation "Amplify$(X, \{X_i\})$" creates a number of identical copies X_i of set X, and then "Discard(X)."

Definition 2-6: Given set $X = \{x_n x_{n-1} \ldots x_2 x_1 | \forall x_d \in \{0,1\}$ for $1 \leq d \leq n\}$ and a bit, x_j, if the value of x_j is equal to one, then the biomolecular *extract* operation creates two new sets, $+(X, x_j^1) = \{x_n x_{n-1} \ldots x_j^1 \ldots x_2 x_1 | \forall x_d \in \{0,1\}$ for $1 \leq d \neq j \leq n\}$ 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 $1 \leq d \neq j \leq n\}$. Otherwise, it produces another two new sets, $+(X, x_j^0) = \{x_n x_{n-1} \ldots x_j^0 \ldots x_2 x_1 | \forall x_d \in \{0,1\}$ for $1 \leq d \neq j \leq n\}$ 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 $1 \leq d \neq j \leq n\}$.

Definition 2-7: Given m sets $X_1 \ldots X_m$, the biomolecular *merge* operation, $\cup(X_1, \ldots, X_m) = X_1 \cup \ldots \cup X_m$.

Definition 2-8: Given set $X = \{x_n x_{n-1} \ldots x_2 x_1 | \forall x_d \in \{0,1\}$ for $1 \leq d \leq n\}$, the biomolecular operation "Detect(X)" returns *true* if $X \neq \emptyset$. Otherwise, it returns *false*.

Definition 2-9: Given set $X = \{x_n x_{n-1} \ldots x_2 x_1 | \forall x_d \in \{0,1\}$ for $1 \leq d \leq n\}$, the biomolecular operation "Read(X)" describes any element in X. Even if X contains many different elements, the biomolecular operation can give an explicit description of exactly one of them.

C. Introduction of Quantum Bits and Quantum Gates

A quantum bit has two computational basis vectors $|0\rangle$ and $|1\rangle$ of the two-dimensional Hilbert space from [8], and corresponds to the classical bit values 0 and 1. A collection of n quantum bits is called a *quantum register* of size n. If the state of a quantum register of size n is arbitrary superposition of the 2^n-dimensional computational basis vectors, then it can be represented as $|\beta\rangle = \sum_{i=0}^{2^n-1} l_i |\rangle$, where each weighted factor $l_i \in \mathbf{C}$ is the so-called probability amplitudes; thus they must satisfy $\sum_{i=0}^{2^n-1} |l_i|^2 = 1$. The time evolution of the states of quantum registers can be modeled by means of quantum gates [8]. From [8], the **Hadamard** gate H is a quantum gate of one quantum bit (a 2×2 matrix), where $H_{1,1} = (1)/(\sqrt{2}), H_{1,2} = (1)/(\sqrt{2}), H_{2,1} = (1)/(\sqrt{2})$, and $H_{2,2} = -(1)/(\sqrt{2})$. The **NOT** gate is a gate with one quantum bit and sets only the target bit to its negation. The **CNOT** (*controlled*-**NOT**) gate is a gate with two quantum bits and flips the target bit if and only if the control bit is equal to one. The *controlled-controlled*-**NOT** (**CCNOT)** gate is a gate with three quantum bits and flips the target bit if and only if the two control bits are both one.

III. Quantum Algorithms for Bio-Molecular Solutions of the Vertex Cover Problem

In this section, the DNA-based algorithm for solving the vertex cover problem of any graph with m edges and n vertices [8] will be introduced. Next, based on Boolean circuits generated from the DNA-based algorithm [8], its corresponding quantum algorithm is presented.

A. All Possible Solutions for the Vertex Cover Problem

From **Definition 2-1**, for any graph G with n vertices and m edges, there are 2^n possible choices including legal and illegal vertex covers in G. Each possible choice corresponds to a subset of vertices in G. Hence, it is supposed that X is a set of 2^n possible choices and X is equal to $\{x_n x_{n-1} \ldots x_2 x_1 | \forall x_d \in \{0,1\}$ for $1 \leq d \leq n\}$. For the sake of presentation, it is assumed that x_d^0 and x_d^1 respectively denote two values "0" and "1" of x_d. For an element $x_n x_{n-1} \ldots x_2 x_1$ in X that is a legal vertex cover, if the value of x_d for $1 \leq d \leq n$ is one, then x_d^1 represents that the dth vertex is within the legal vertex cover. Otherwise x_d^0 represents that the dth vertex is not within the legal vertex cover. **Definition 3-1** is used to denote how each element in X is represented as a unique *computational basis vector* with 2^n-tuples of binary numbers.

Definition 3-1: The jth element in X can be represented as a unique *computational basis vector* $|u_j\rangle = \lfloor u_{1,1} \quad u_{1,2} \quad \cdots \quad u_{1,2^n} \rfloor_{1 \times 2^n}^T$, where $u_{1,j} = 1$ and $\forall u_{1,h} = 0$ for $1 \leq h \neq j \leq 2^n$.

B. Computational State Space of Molecular Solutions for the Vertex Cover Problem

For solving the vertex cover problem of a graph with m edges and n vertices, the following biomolecular algorithm can be applied to create all of the 2^n possible choices. A set X_0 is an empty set and is regarded as the input set of the following DNA-based algorithm. The second parameter n in **ComputationalStateSpace**(X_0, n) is used to represent the number of vertices. It is assumed that tubes Y_1 and Y_2 in **ComputationalStateSpace**(X_0, n) are initially empty tubes.

Procedure ComputationalStateSpace(X_0, n)

(0a) Append-Tail(Y_1, x_n^1).

(0b) Append-Tail(Y_2, x_n^0).

(0c) $X_0 = \cup(Y_1, Y_2)$.

(1) **For** $d = n - 1$ **downto** 1

(1a) Amplify(X_0, Y_1, Y_2).

(1b) Append-Tail(Y_1, x_d^1).

(1c) Append-Tail(Y_2, x_d^0).

(1d) $X_0 = \cup(Y_1, Y_2)$.

End For

End Procedure

Lemma 3-1: For solving the vertex cover problem of a graph with m edges and n vertices, 2^n possible choices are created from the DNA-based algorithm **ComputationalStateSpace**(X_0, n), and the set of the corresponding computational state vectors of 2^n possible choices forms an orthonormal basis of a 2^n dimensional Hilbert space (a complex vector space, $\boldsymbol{C}^{2^n}$).

Proof: Each execution of Step (0a) and Step (0b), respectively, append the value "1" for x_n as the first bit of every element in a set Y_1 and the value "0" for x_n as the first bit of every element in a set Y_2. That gives that $Y_1 = \{x_n^1\}$ and $Y_2 = \{x_n^0\}$. Next, each execution of Step (0c) creates the set union for the two sets Y_1 and Y_2 so that $X_0 = Y_1 \cup Y_2 = \{x_n^1, x_n^0\}$, and $Y_1 = \emptyset$ and $Y_2 = \emptyset$.

Each execution of Step (1a) creates two identical copies, Y_1 and Y_2, of set X_0, and $X_0 = \emptyset$. Each execution of Step (1b) then appends the value "1" for x_d onto the end of $x_n \ldots x_{d+1}$ for every element in Y_1. Similarly, each execution of Step (1c) also appends the value "0" for x_d onto the end of $x_n \ldots x_{d+1}$ for every element in Y_2. Next, each execution of Step (1d) creates the set union for the two sets Y_1 and Y_2 so that $X_0 = Y_1 \cup Y_2$, and $Y_1 = \emptyset$ and $Y_2 = \emptyset$. After repeating Steps (1a) through (1d), $X_0 = \{x_n x_{n-1} \ldots x_2 x_1 | \forall x_d \in \{0, 1\}$ for $1 \leq d \leq n\}$ is obtained. This implies that 2^n possible choices are produced. From **Definition 3-1**, {- $[1\ \ 0\ \ \cdots\ \ 0]^T_{1\times 2^n} [0\ \ 1\ \ \cdots\ \ 0]^T_{1\times 2^n} \cdots [0\ \ 0\ \ \cdots\ \ 1]^T_{1\times 2^n}\}$ is the set of the corresponding computational basis vectors for each element in the tube X_0, and from [8] its span is $\boldsymbol{C}^{2^n}$. This is to say that it forms an orthonormal basis of a 2^n dimensional Hilbert space. ■

C. *Mathematical Solutions of Computational State Space of Molecular Solutions for the Vertex Cover Problem*

It is assumed that a quantum register of n bits, $(\otimes_{q=n}^{1} |x_q^0\rangle)$, is used to initialize a system that has $P = 2^n$ states which are labeled as $Q_0, Q_1, Q_2, \ldots, Q_{P-1}$, where each state Q_k for $0 \leq k \leq 2^n - 1$ corresponds to the kth possible molecular solution. For labeling the amplitude of the answer(s) among 2^n states, one Hadamard gate, H, is used to operate $(|1\rangle)$ and the quantum state vector $((1)/(\sqrt{2})(|0\rangle - |1\rangle))$ is obtained. It is assumed that the initial quantum state vector $(|\theta_0\rangle)$ is $(\otimes_{q=n}^{1} |x_q^0\rangle)$. The system that has P $= 2^n$ states which are labeled as $Q_0, Q_1, Q_2, \ldots, Q_{P-1}$ can be initialized to the distribution: $((1)/(\sqrt{2^n}), (1)/(\sqrt{2^n}), (1)/(\sqrt{2^n}), \ldots, (1)/(\sqrt{2^n}))$, i.e., there is the same amplitude in each of the 2^n states. This distribution can be obtained by means of n Hadamard gates operating the initial quantum state vector $(|\theta_0\rangle)$.

D. *Molecular Solutions of Finding Legal Vertex Covers Among 2^n Possible Choices*

It is assumed that the kth edge, $e_k = (v_i, v_j)$, in G to $1 \leq k \leq m$ and bits x_i and x_j represent vertices v_i and v_j, respectively. Because a legal vertex cover consists of at least one vertex from the kth edge in G for $1 \leq k \leq m$, the requested condition can be represented as a Boolean formula of the form

$$F(x_n, x_{n-1}, \ldots, x_2, x_1) = C_1 \wedge C_2 \ldots C_{m-1} \wedge C_m, \quad (3\text{-}1)$$

where each C_j for $1 \leq j \leq m$ is a clause with the form $x_i \vee x_j$. Therefore, the question is to find choices among 2^n possible choices that satisfy it.

The following biomolecular algorithm can be used to find legal vertex covers and to remove illegal vertex covers among 2^n possible choices. 2^n possible molecular solutions in a set X_0 are produced by the DNA-based algorithm, **ComputationalStateSpace**(X_0, n), and the set X_0 is regarded as the input set of the following DNA-based algorithm. The second parameter n in **FindingLegalVertexCover**(X_0, n, m) is used to represent the number of vertices, and the third parameter m in **FindingLegalVertexCover**(X_0, n, m) is applied to represent the number of edges.

Procedure FindingLegalVertexCover(X_0, n, m)

(1) **For** each edge, $e_k = (v_i, v_j)$, in G to $1 \leq k \leq m$ and bits x_i and x_j respectively represent vertices v_i and v_j.

(1a) $\theta^1 = +(X_0, x_i^1)$ and $\theta^3 = -(X_0, x_i^1)$.

(1b) $\theta^2 = +(\theta^3, x_j^1)$ and $\theta^4 = -(\theta^3, x_j^1)$.

(1c) $X_0 = \cup(\theta^1, \theta^2)$.

(1d) Discard(θ^4).

End For

End Procedure

Lemma 3-2: For the vertex cover problem of a graph G with m edges and n vertices, the DNA-based algorithm **FindingLegalVertexCover**(X_0, n, m) can be applied to find the legal vertex covers and to remove illegal vertex covers among 2^n possible choices created by **ComputationalStateSpace**(X_0, n).

Proof: On each execution of Step (1a) and Step (1b), tube θ^1 contains DNA strands that have $x_i = 1$, tube θ^2 contains DNA strands that have $x_j = 1$ and $x_i = 0$, tube θ^4 contains DNA strands that have $x_j = 0$ and $x_i = 0$, tube $X_0 = \emptyset$ and tube $\theta^3 = \emptyset$. This implies that molecular solutions in tubes θ^1 and θ^2 at least contain one of two vertices in the kth edge and are legal vertex covers, and molecular solutions in tube θ^4 do not contain any vertex in the kth edge and are not legal vertex covers. Then, on each execution of Step (1c) and Step (1d), DNA strands in tube X_0 at least encode one vertex in the kth edge, tube $\theta^1 = \emptyset$, tube $\theta^2 = \emptyset$, and illegal vertex covers in tube θ^4 are removed so that tube $\theta^4 = \emptyset$. After repeating to execute Steps (1a) through (1d), tube X_0 consists of DNA strands that satisfy each formula with the form $x_i \vee x_j$ for the kth edge in G for $1 \leq k \leq m$. ■

E. Mathematical Solutions of Molecular Solutions of Legal Vertex Covers Among 2^n Possible Choices

From [8], the operation **OR** can be implemented by two NOT gates and one **CCNOT** gate with the target bit that is initially set to state $|1\rangle$. The operation **AND** can be also implemented by one **CCNOT** gate with the target bit that is initially set to state $|0\rangle$. For implementing the function of the Boolean formula (3-1) with $F(x_n, x_{n-1}, \ldots, x_2, x_1) = C_1 \wedge C_2 \ldots C_{m-1} \wedge C_m$, two auxiliary quantum registers $|r_m^1 \cdots r_1^1\rangle$ and $|c_m^0 c_{m-1}^0 \cdots c_1^0 c_0^1\rangle$ are needed. Because the previous clause of the first clause does not exist, the first quantum bit of the third quantum register is $|c_0^1\rangle$. The kth quantum bit $|r_k\rangle$ in the second quantum register is employed to store the result for the kth clause with the form $x_i \vee x_j$. The kth quantum bit $|c_k\rangle$ in the third quantum register is employed to store the result to the current clause (the kth clause) and the previous clause (the $(k-1)$th clause). The $(m+1)$th quantum bit $|c_m\rangle$ in the third register is employed to store the final result for all of the clauses.

Lemma 3-3: To solve the vertex cover problem of any graph G with n vertices and m edges, Boolean circuits generated from the DNA-based algorithm **FindingLegalVertexCover**(X_0, n, m) and to judge which among 2^n possible choices are legal vertex covers and which are not answers can be implemented by quantum evaluating circuits (**QEC**) that are made of **NOT** gates and **CCNOT** gates.

Proof: Because Boolean circuits generated from the DNA-based algorithm **FindingLegalVertexCover**(X_0, n, m) is actually to implement the function of the Boolean formula (3-1) with $F(x_n, x_{n-1}, \ldots, x_2, x_1) = C_1 \wedge C_2 \ldots C_{m-1} \wedge C_m$. For implementing it, two auxiliary quantum registers $|r_m^1 \cdots r_1^1\rangle$ and $|c_m^0 c_{m-1}^0 \cdots c_1^0 c_0^1\rangle$ are needed, m **OR** operations through the relation $(|r_k^1 \oplus \bar{x}_i \cdot \bar{x}_j\rangle)$ for $1 \leq i$ and $j \leq n$ and $1 \leq k \leq m$ are completed, and m **AND** operations through the relation $(|c_k^0 \oplus c_{k-1} \cdot r_k\rangle)$ for $1 \leq k \leq m$ are completed in [8]. This is to say that m **OR** operations and m **AND** operations are all implemented by means of **CCNOT** gates and **NOT** gates. ■

Lemma 3-4: Mathematical solutions of molecular solutions of legal vertex covers created by the DNA-based algorithm **FindingLegalVertexCover**(X_0, n, m) are a unit vector in a finite-dimensional Hilbert space (a complex vector space, C^{2^n}).

Proof: From **Lemma 3-3**, Boolean circuits generated from the DNA-based algorithm **FindingLegalVertexCover**(X_0, n, m) can be implemented by means of quantum evaluating circuits (**QEC**) that are made of **NOT** gates and **CCNOT** gates. Because the new quantum state vector is still a unit vector, hence, it is at once inferred that their mathematical solutions are a unit vector in a finite-dimensional Hilbert space (a complex vector space, C^{2^n}). ■

F. Molecular Solutions of Finding a Minimum-Sized Vertex Cover Among Legal Vertex Covers

The following biomolecular algorithm can be used to find a *minimum-sized* vertex cover among legal vertex covers. Molecular solutions of legal vertex covers in a set X_0 are produced by the DNA-based algorithm, **FindingLegalVertexCover**(X_0, n, m), and the set X_0 is regarded as the input set of the following DNA-based algorithm. In **FindingMinumumSizedVertexCover**(X_0, n, m), the second parameter n is used to represent the number of vertices, and the third parameter m is applied to represent the number of edges. In **FindingMinumumSizedVertexCover**(X_0, n, m), each set X_k for $1 \leq k \leq n$ is initialized to an empty set, and each set X_k^{ON} for $1 \leq k \leq n$ is also initialized to an empty set.

Procedure FindingMinumumSizedVertexCover(X_0, n, m)

(1) **For** $i = 0$ **to** $n - 1$

  (2) **For** $j = i$ **down to** 0

    (2a) $X_{j+1}^{ON} = +(X_j, x_{i+1}^1)$ and $X_j = -(X_j, x_{i+1}^1)$.

    (2b) $X_{j+1} = \cup(X_{j+1}, X_{j+1}^{ON})$.

  End For

End For

End Procedure

Lemma 3-5: The DNA-based algorithm, **FindingMinumumSizedVertexCover**(X_0, n, m), for the vertex cover problem of a graph G with m edges and n vertices can be used to find a *minimum-sized* vertex cover among legal vertex covers yielded by **FindingLegalVertexCover**(X_0, n, m).

Proof: At the iteration (0, 0) in the two-level nested loop, on the first execution of Step (2a) and Step (2b), the influence of x_1 for the number of ones is to record one one in X_1 and also to record zero ones in X_0 and $X_1^{ON} = \emptyset$. Next, at the iteration (1, 1) in the two-level nested loop, on the *second* execution of Step (2a) and Step (2b), the influence of x_2 for the number of ones is to record two ones in X_2 and to record one one in X_1. Next, at the iteration (1, 0) in the two-level nested loop, on the third execution of Step (2a) and Step (2b), the influence of x_2 for the number of ones is to record one one in X_1 and also to record zero ones in X_0. Next, from the iteration (2, 2) through the iteration $(n-1, 0)$ in the two-level nested loop, similar processing is applied to compute the influence of x_3 through x_n for the number of ones. Hence, after each operation is completed, those combinations in X_i for $0 \leq i \leq n$ have i ones. ■

G. Mathematical Solutions of Molecular Solutions of Minimum-Sized Vertex Covers

For performing Boolean circuits generated from Steps (2a) and (2b) at the same iteration in **FindingMinumumSizedVertexCover**(X_0, n, m), auxiliary quantum bits for $0 \leq i \leq n-1$ and $0 \leq j \leq i |z_{i+1,j}\rangle, |z_{i+1,i+1}\rangle, |g_{i,j,0}\rangle, |f_{i,j,0}\rangle, |h_{i,j,i-j+1}\rangle, |h_{i,j,0}\rangle$, and $|z_{0,0}\rangle$ are needed. For $0 \leq i \leq n-1$ and $0 \leq j \leq i$, each quantum bit in $|z_{i+1,j}\rangle, |z_{i+1,i+1}\rangle, |g_{i,j,0}\rangle, |f_{i,j,0}\rangle$, and $|h_{i,j,i-j+1}\rangle$ is initially prepared in state $|0\rangle$, and each quantum bit in $|h_{i,j,0}\rangle$ and $|z_{0,0}\rangle$ is initially prepared in state $|1\rangle$. Assume that for $0 \leq i \leq n-1$ and $0 \leq j \leq i, |z_{i+1,j}\rangle$ is applied to record the status of tube (set) X_{j+1} that has $(j+1)$ ones, and $|z_{i+1,j}\rangle$ is used to record the status of tube (set) X_j that has j ones after the influence of x_{i+1} to the number of ones is figured out from the loop iteration (i, j) in the two-level nested loop in **FindingMinumumSizedVertexCover**(X_0, n, m). Boolean circuits generated from Steps (2a) and (2b) at the same iteration in **FindingMinumumSizedVertexCover**(X_0, n, m) can be represented as a Boolean formula of the form

$$|z_{i+1,j+1}\rangle = |z_{i+1,j+1}\rangle \oplus (|c_m\rangle \wedge (x_{i+1} \wedge |z_{i,j}\rangle \wedge (\wedge_{k=j+2}^{i+1} |\bar{z}_{i+1,k}\rangle))) \text{ and } |z_{i+1,j}\rangle = |z_{i+1,j}\rangle \oplus (|c_m\rangle \wedge \bar{x}_{i+1} \wedge |z_{i,j}\rangle). \quad (3\text{-}2)$$

Lemma 3-6: Boolean circuits produced from Steps (2a) and (2b) at the same iteration (i, j) in the two-level nested loop in **FindingMinumumSizedVertexCover**(X_0, n, m) can be implemented by means of quantum circuits called **FMNO** (the abbreviation of finding the minimum number of ones) that are made of **CCNOT** gates and **NOT** gates.

Proof: A Boolean formula of the form (3-2) with $|z_{i+1,j+1}\rangle = |z_{i+1,j+1}\rangle \oplus (|c_m\rangle \wedge (x_{i+1} \wedge |z_{i,j}\rangle \wedge (\wedge_{k=j+2}^{i+1}|\bar{z}_{i+1,k}\rangle)))$ and $|z_{i+1,j}\rangle = |z_{i+1,j}\rangle \oplus (|c_m\rangle \wedge \bar{x}_{i+1} \wedge |z_{i,j}\rangle)$ is actually Boolean circuits generated from Steps (2a) and (2b) at the same iteration in **FindingMinumumSizedVertexCover**(X_0, n, m), so for $0 \le i \le n-1$ and $0 \le j \le i$, each bit in $|z_{i+1,j+1}\rangle$ is an auxiliary quantum bit and is applied to store the result of performing the *first* condition in (3-2). Therefore, this step requires computing the **AND** operations through the relation $h_{i,j,a} \leftarrow h_{i,j,a} \oplus (h_{i,j,a-1} \cdot \bar{z}_{i+1,k})$, and for $1 \le a \le i-j$ and $j+2 \le k \le i+1$, the relation $h_{i,j,i-j+1} \leftarrow h_{i,j,i-j+1} \oplus (h_{i,j,i-j} \cdot z_{i,j})$ and the relation $f_{i,j,0} \leftarrow f_{i,j,0} \oplus (h_{i,j,i-j+1} \cdot x_{i+1})$. Next, it also requires figuring out the **CCNOT** operation through the relation $z_{i+1,j+1} \leftarrow z_{i+1,j+1} \oplus (c_m \cdot f_{i,j,0})$. It then requires subsequently computing the **NOT** operations on $\bar{z}_{i+1,k}$ for $j+2 \le k \le i+1$ to restore each quantum bit $\bar{z}_{i+1,k}$ to its previous state. This enables the reuse of $\bar{z}_{i+1,k}$ for $j+2 \le k \le i+1$.

For $0 \le i \le n-1$ and $0 \le j \le i$, each bit in $|z_{i+1,j}\rangle$ is also an auxiliary quantum bit, and is applied to store the result of performing the function of the *second* condition in (3-2). This step requires computing the **NOT** operation on $x_{i+1}(\bar{x}_{i+1})$, the **AND** operation, and the **CCNOT** operation through the relation $g_{i,j,0} \leftarrow g_{i,j,0} \oplus (z_{i,j} \cdot \bar{x}_{i+1})$ and the relation $z_{i+1,j} \leftarrow z_{i+1,j} \oplus (c_m \cdot g_{i,j,0})$. Next, it requires subsequently figuring out the **NOT** operations on $|x_n \cdots x_1\rangle$ to restore each quantum bit in $|x_n \cdots x_1\rangle$ to its superposition state. This enables us to preserve the superposition in $|x_n \cdots x_1\rangle$ and to reuse the superposition in $|x_n \cdots x_1\rangle$. Therefore, it is at once inferred that Boolean circuits produced from Steps (2a) and (2b) at the same iteration (i, j) in the two-level nested loop in **FindingMinumumSizedVertexCover**(X_0, n, m) can be implemented by means of quantum circuits called **FMNO** (the abbreviation of finding the minimum number of ones) that are made of **CCNOT** gates and **NOT** gates.

H. Reading Molecular Solutions of Minimum-Sized Vertex Covers

The following biomolecular algorithm can be used to read molecular solutions of a *minimum-sized* vertex cover among minimum-sized vertex covers. Molecular solutions of minimum-sized vertex covers in a set X_0 are produced by the DNA-based algorithm, **FindingMinumumSizedVertexCover**(X_0, n, m), and the set X_0 is regarded as the input set of the following DNA-based algorithm. In **ReadingAnswer**(X_0, n, m), the second parameter n is used to represent the number of vertices, and the third parameter m is applied to represent the number of edges. In **ReadingAnswer**(X_0, n, m), tubes X_1 through X_n are all generated by the DNA-based algorithm **FindingMinumumSizedVertexCover**(X_0, n, m), and tube X_c includes those DNA strands encoding a vertex cover with c vertices for $1 \le c \le n$.

Procedure ReadingAnswer(X_0, n, m)

(1) **For** $c = 1$ **to** n

(1a) **If** $(\text{detect}(X_c))$ **then**

(1b) Read(X_c) and terminate the algorithm.

EndIf

EndFor

End Procedure

Lemma 3-7: For the vertex cover problem of a graph G with m edges and n vertices, the DNA-based algorithm, **ReadingAnswer**(X_0, n, m), can be employed to read molecular solutions of a minimum-sized vertex cover among minimum-sized vertex covers created by **FindingMinumumSizedVertexCover**(X_0, n, m).

Proof: On each execution of Step (1a), a "true" is returned if there are DNA strands in tube X_c. This indicates that the number of vertices for a minimum-sized vertex cover is the value of the loop index variable, c. Next, on each execution of Step (1b), the answer of a minimum-sized vertex cover is read and the algorithm is terminated. ■

I. Mathematical Solutions for Reading Molecular Solutions of Minimum-Sized Vertex Covers

Grover's operator in [8] is used to increase exponentially the amplitude or probability of finding the answer(s), and is defined by matrix G as follows: $G_{i,j} = ((2)/(2^n))$ if $i \ne j$ and $G_{i,i} = (-1 + (2)/(2^n))$. **Algorithm 3-1** is applied to complete one operation detect(X_c) and one operation Read(X_c) in Step (1a) and Step (1b) in **ReadingAnswer**(X_0, n, m). The notations used in **Algorithm 3-1** below have been denoted in previous subsections. The first parameter w in **Algorithm 3-1** is used to represent the minimum size of vertices among legal answers, and its value is passed from the execution of Step (1a) in **Algorithm 3-2** in the next subsection.

Algorithm 3-1 (w): Mathematical solutions of reading molecular solutions of minimum-sized vertex covers for any graph G with m edges and n vertices.

(1) A unitary operator $(H)(H^{\otimes n})$ is applied to operate an initial quantum state vector $|\Phi\rangle$ with $(|1\rangle) \otimes (\otimes_{b=n}^{1} |x_b^0\rangle)$, and 2^n possible choices of n bits are obtained: $((|0\rangle - |1\rangle)/(\sqrt{2})) \otimes (|\varphi_{0,0}\rangle) = ((|0\rangle - |1\rangle)/(\sqrt{2}))(1)/(\sqrt{2^n})(\otimes_{b=n}^{1}(|x_b^0\rangle + |x_b^1\rangle))$.

(2) For labeling legal vertex covers, a quantum circuit, $(I_{2\times 2})$ (**EQC**), is used to

operate the state vector $((|0\rangle - |1\rangle)/(\sqrt{2})) \otimes ((\otimes_{k=m}^{1}|c_k^0\rangle)(|c_0^1\rangle)(\otimes_{k=m}^{1}|r_k^1\rangle)) \otimes (|\varphi_{0,0}\rangle)$, and the new state vector is obtained: $((|0\rangle - |1\rangle)/(\sqrt{2})) \otimes (|\varphi_{1,0}\rangle)= ((|0\rangle - |1\rangle)/(\sqrt{2}))(1)/(\sqrt{2^n})(\sum_{X=0}^{2^n-1}((\otimes_{k=m}^{1}|c_k^0 \oplus (c_{k-1} \cdot r_k)\rangle)(|c_0^1\rangle)(\otimes_{k=m}^{1}|r_k^1\rangle \oplus (\overline{x_i} \cdot \overline{x_j})\rangle)(|X\rangle)))$.

(3) **For** $i = 0$ **to** $n-1$

(4) **For** $j = i$ **down to** 0

(4a) A quantum circuit, $(I_{2\times 2})$ (**FMNO**), is applied to calculate the number of vertices among the legal vertex covers and is also used to operate the state vector $((|0\rangle - |1\rangle)/(\sqrt{2})) \otimes ((\otimes_{i=n}^{1} \otimes_{j=i}^{0} |z_{i,j}^0\rangle)(|z_{0,0}^1\rangle)(\otimes_{i=n-1}^{0} \otimes_{j=i}^{0} |g_{i,j,0}^0\rangle)(\otimes_{i=n-1}^{0} \otimes_{j=i}^{0} |f_{i,j,0}^0\rangle)(\otimes_{i=n-1}^{0} \otimes_{j=i}^{0} ((\otimes_{a=i-j+1}^{1}|h_{i,j,a}^0\rangle) \otimes (|h_{i,j,0}^1\rangle))))\otimes(|\varphi_{1+(\sum_{\theta_1=0}^{i-1}(\theta_1+1))+(i-j),0}\rangle)$. Because Step (4a) is embedded in the only loop, after repeating to execute the quantum circuit, $(I_{2\times 2})$(**FMNO**), the resulting state vector, $|\varphi_{1+(n\times(n+1))/(2),0}\rangle = ((|0\rangle - |1\rangle)/(\sqrt{2}))(1)/(\sqrt{2^n})(\sum_{X=0}^{2^n-1}((\otimes_{i=n}^{1} \otimes_{j=i}^{0} |z_{i,j}\rangle)(|z_{0,0}^1\rangle)(\otimes_{i=n-1}^{0} \otimes_{j=i}^{0} |g_{i,j,0}\rangle)(\otimes_{i=n-1}^{0} \otimes_{j=i}^{0} |f_{i,j,0}\rangle)(\otimes_{i=n-1}^{0} \otimes_{j=i}^{0} ((\otimes_{a=i-j+1}^{1}|h_{i,j,a}\rangle) \otimes(|h_{i,j,0}^1\rangle))))(\otimes_{k=m}^{1}|c_k\rangle)(|c_0^1\rangle)(\otimes_{k=m}^{1}|r_k\rangle)(|X\rangle)))$ is obtained in which the number of vertices in each legal vertex cover is calculated.

End For

End For

(5) A ***CNOT*** gate $((|0\rangle - |1\rangle)/(\sqrt{2}) \oplus z_{n,w})$ is used to label the legal vertex cover(s) with the minimum number of vertices in the quantum state $|\varphi_{1+(n\times(n+1))/(2),0}\rangle$, and the resulting new quantum state vector is obtained: $|\varphi_{1+(n\times(n+1))/(2)+1,0}\rangle = (1)/(\sqrt{2^n}) \times (-1)^{z_{n,w}}(|\varphi_{1+(n\times(n+1))/(2),0}\rangle)$.

(6) Since quantum operations are reversible by nature, the auxiliary quantum bits can be restored to their initial states by reversing all these operations finished by Steps (4a) and (2).

(7) Apply Grover's operator in **Grover's algorithm** to the quantum state vector generated in Step (6).

(8) At most repeat executing from Step (2) to Step (7) $(O(\sqrt{(2^n)/(R)}))$ times, where the value of R is the number of solutions and can be efficiently computed from the quantum counting algorithm [8].

(9) The answer is obtained with a successful probability of at least $(1)/(2)$ after a measurement is finished.

End Algorithm

Lemma 3-8: The output of **Algorithm 3-1** is mathematical solutions for reading molecular solutions of the minimum-sized vertex covers for any graph G with m edges and n vertices.

Proof: From the execution of Step (1), 2^n possible vertex covers in the state vector $((|0\rangle - |1\rangle)/(\sqrt{2})) \otimes (|\varphi_{0,0}\rangle)$ is obtained. This implies that the function of the DNA-based algorithm **ComputationalStateSpace**(X_0, n) can be implemented by Step (1) in **Algorithm 3-1**. Next, from the execution of Step (2), *legal* vertex covers in the resulting state vector $((|0\rangle - |1\rangle)/(\sqrt{2})) \otimes (|\varphi_{1,0}\rangle)$ are found. This indicates that the function of the DNA-based algorithm **FindingLegalVertexCover**(X_0, n, m) can be implemented by Step (2) in **Algorithm 3-1**. Next, after repeating to execute Step (4a), the resulting state vector $|\varphi_{1+(n\times(n+1))/(2),0}\rangle$ is obtained in which the number of vertices in each legal vertex cover is calculated. This implies that the function of the DNA-based algorithm **FindingMinumumSizedVertexCover**(X_0, n, m) can be implemented by Step (4a) in **Algorithm 3-1**.

Next, one **CNOT** gate, $((|0\rangle - |1\rangle)/(\sqrt{2}) \oplus z_{n,w})$, in the execution of Step (5) is used to perform the oracle work (in the language of **Grover's algorithm**), that is, the target state labeling preceding Grover's searching step. The resulting state vector $|\varphi_{1+(n\times(n+1))/(2)+1,0}\rangle$ contains the part of the answer with phase -1 and the other part with phase $+1$.

Next, the execution of Step (6) is used to reverse all those operations completed by Steps (4a) and (2) so that the auxiliary quantum bits can be restored to their initial states and then they can be repeated for safe use. Next, on the execution of Step (7), it applies Grover's operator to perform the task of increasing the probability of success in measuring the answer. From Step (8), after repeating the execution of Steps (2) through (7) $(O(\sqrt{(2^n)/(R)}))$ times, a maximum successful probability is generated. Next, from the execution of Step (9), a measurement is used to obtain the answer(s) and the answer(s) is/are returned to **Algorithm 3-2**. Since the result generated by each step in **Algorithm 3-1** is a unit vector in the finite-dimensional Hilbert space, therefore, it is at once inferred that the output of **Algorithm 3-1** is mathematical solutions of reading molecular solutions of the minimum-sized vertex covers for any graph G with m edges and n vertices. ■

J. Measuring Answers to Mathematical Solutions for Reading Molecular Solutions of Minimum-Sized Vertex Covers for Solving the Vertex Cover Problem of Any Graph G With m Edges and n Vertices

The following algorithm is applied to solve an instance of the vertex cover problem of any graph G with m edges and n vertices. The notations used in **Algorithm 3-2** below have been denoted in previous subsections.

Algorithm 3-2: Measuring answers to mathematical solutions of reading molecular solutions of minimum-sized vertex covers for solving an instance of the vertex cover problem of any Graph G with m edges and n vertices.

(1) **For** $w = 1$ **to** n

(1a) Call **Algorithm 3-1**(w).

(1b) **If** the answer is obtained from the wth execution of Step (1a) **then**

(1c) Terminate **Algorithm 3-2**.

End If

End For

End Algorithm

Lemma 3-9: **Algorithm 3-2** is applied to obtain the answer(s) to solve an instance of the vertex cover problem of any graph G with m edges and n vertices after measuring mathematical solutions of reading molecular solutions of minimum-sized vertex covers.

Proof: In each execution of Step (1a) in **Algorithm 3-2**, the answer(s) from **Algorithm 3-1** is/are returned to **Algorithm 3-2**. That is to say that it is shown that mathematical solutions of molecular solutions for finding the minimum-sized vertex covers are a unit vector in the finite-dimensional Hilbert space and mathematical solutions of reading molecular solutions of the minimum-sized vertex covers are still a unit vector in the finite-dimensional Hilbert space. Next, in each execution of Step (1b) in **Algorithm 3-2**, if the answer is found from the wth execution of Step (1a) in **Algorithm 3-2**, then the wth execution of Step (1c) in **Algorithm 3-2** is applied to terminate **Algorithm 3-2**. Otherwise, Steps (1a) through (1c) are executed until the answer is found. Therefore, it is at once inferred that **Algorithm 3-2** can be applied to obtain the answer(s) to solve an instance of the vertex cover problem of any graph G with m edges and n vertices after measuring mathematical solutions of reading molecular solutions of minimum-sized vertex covers. ■

IV. Experimental Results

For a graph $G = \{\{v_1\}, \{(v_1, v_1)\}\}$, its minimum vertex cover is actually $\{v_1\}$. For finding the answer, an quantum operator (**CNOT**) $\otimes$ (**CNOT**) $\otimes (H)$ is used to operate $(|z^0_{1,1}\rangle \otimes |\boldsymbol{c}^0_1\rangle \otimes |x^0_1\rangle)$ and the new state vector $|W_1\rangle = (1)/(\sqrt{2})(|z^0_{1,1}\rangle|c^0_1\rangle|x^0_1\rangle + |z^1_{1,1}\rangle|c^1_1\rangle|x^1_1\rangle)$ is obtained. Our experiment is carried out on a Varian INOVA 600 NMR spectrometer. The sample is $^{13}C-$ *labelled* alanine with formula $^{13}_{1}$ CH_3 $-^{13}_{2}$ $CH(NH_2)$ $-^{13}_{3}$ $COOH$, where the three carbons $^{13}_{1}C,^{13}_{2}C$, and $^{13}_{3}C$ correspond to the quantum bits I_1, I_2, and I_3, respectively. The J-coupling constants are $J_{12} = 34.79$ Hz, $J_{23} = 54.01$ Hz, and $J_{13} = 1.20$ Hz. Soft pulses are used to achieve the selective excitation. If the algorithm works correctly, the detection of the nuclear spins of the three ^{13}C in $|0\rangle$ should correspond to the absorption peaks in the **NMR** spectra at 2500 Hz, 7620 Hz, and 11 955 Hz, respectively.

The states of the input quantum bits can be written in the form of the product operations as follows: $\boldsymbol{E} + \boldsymbol{I_{1Z}} + \boldsymbol{I_{2Z}} + \boldsymbol{I_{3Z}} + \boldsymbol{2I_{1Z}I_{2Z}} + \boldsymbol{2I_{2Z}I_{3Z}} + \boldsymbol{2I_{1Z}I_{3Z}} + \boldsymbol{4I_{1Z}I_{2Z}I_{3Z}}$, where $\boldsymbol{E}$ is the unity operator with the form of $\boldsymbol{E} = (\begin{smallmatrix}1 & 0\\ 0 & 1\end{smallmatrix})$, and $\boldsymbol{I_{iZ}} = (1)/(2)\boldsymbol{\sigma_z}$, with $i = 1, 2$ and 3, being the ith spin angular momentum operator in the $Z-$ direction, and $\boldsymbol{\sigma_z}$ is the Pauli matrix $\boldsymbol{\sigma_z} = (\begin{smallmatrix}1 & 0\\ 0 & -1\end{smallmatrix})$. Next, the Hadamard gate can be achieved by a single $(\boldsymbol{\pi}/2)$ pulse with phase x. The **CNOT** gate can be implemented by NMR pulses as follows: $[\pi/2]^2_y \rightarrow (1/4\boldsymbol{J}) \rightarrow [\pi]^{1,2}_x \rightarrow (1/4\boldsymbol{J}) \rightarrow [\pi]^{1,2}_x \rightarrow [\pi/2]^2_x$, where the flip angle of the pulse and the time of delay are written in square brackets and in round brackets, respectively. The subscripts are the phases (i.e., along the x or y axis) of the pulse, and the superscripts are the nuclei to which the pulses are applied. Then we could obtain the total pulse sequence by connecting and optimizing the aforesaid pulses according to the quantum circuit. Next, a readout pulse is applied to each quantum bit to obtain the spectra.

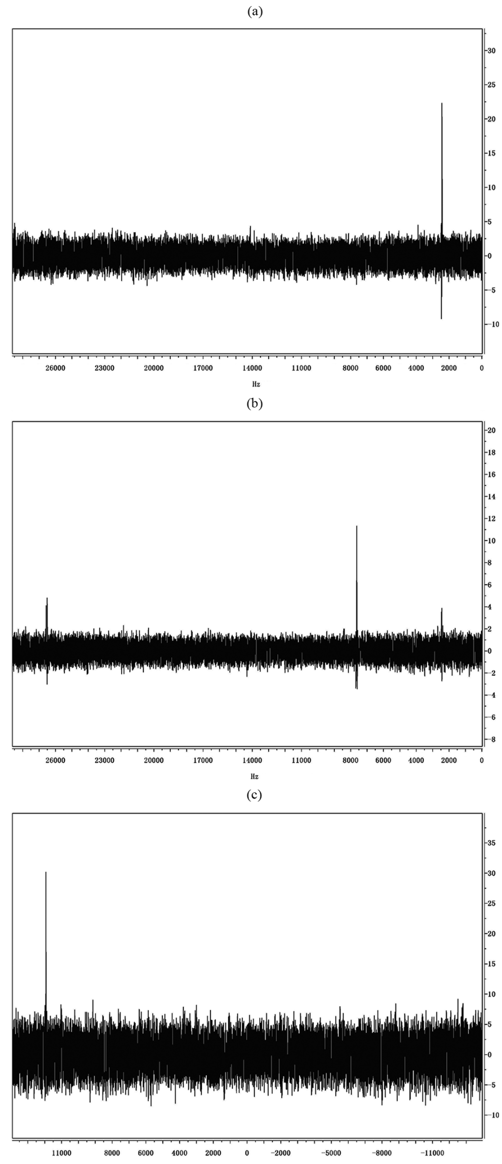

Fig. 1. Experimental spectra (a)–(c) of the three-quantum- bit (^{13}C) solution for the vertex-cover problem after the readout on the first, second and third quantum bits, respectively, where the phases of the reference of the ^{13}C **NMR** spectra for a thermal equilibrium have been adjusted to be in absorption (i.e., positive), and then the same phase corrections are used to determine the absolute phases of the experimental spectra of ^{13}C. These absorption peaks of the ^{13}C **NMR** spectra represent the three quantum bits to be in the states $|0\rangle$, respectively, after the disentangling operations are performed.

In our case, the final state was $(|000\rangle_{123} + |111\rangle_{123})/\sqrt{2}$ which means the three quantum bits are entangled. As the readout by NMR is a weak measurement, we have no state collapse after the measurement. Besides, only single quantum coherence can be detected in NMR. As a result, we have to employ some additional operations to disentangle them for detecting the output state $(|000\rangle_{123} + |111\rangle_{123})/\sqrt{2}$. For this end, we apply a **CNOT** gate on the second and first quantum bits to get the state $(|000\rangle_{123} + |011\rangle_{123})/\sqrt{2}$. The second quantum bit is the control bit and the first is the target bit. Then the first quantum bit can be read out by a single $(\boldsymbol{\pi}/2)$ pulse along the x-axis, as shown in Fig. 1(a) where the horizontal axis is for frequency and the vertical axis is for signal strength. The peak appears at 2492 Hz meaning that ${}^{13}_{1}C$ is detected to be in $|0\rangle$. Similar steps applied to the second and third quantum bits, respectively, result in peaks at 7615 Hz in Fig. 1(b) and 11 950 Hz in Fig. 1(c).

V. Conclusion

From **Lemma 3-8** and **Lemma 3-9**, the quantum algorithm using ideas (Boolean circuits) from DNA computing for solving the vertex cover problem of any graph G with m edges and n vertices is the *optimal* quantum algorithm. The number of quantum bits for solving it is the successful key for a quantum system of a real-world situation (for example, **NMR** technology). From **Lemma 3-8**, the space consumption for the worst, average and best case is the same, and is $((2 \times m + 3) + ((n^3 + 15 \times n^2 + 26 \times n)/(6)))$ quantum bits.

References

[1] R. P. Feynman, *Minaturization*, D. H. Gilbert, Ed. New York: Reinhold, 1961, pp. 282–296.

[2] R. Feynman, "Simulating physics with computers," *Int. J. Theor. Phys.*, vol. 21, no. 6/7, pp. 467–488, 1982.

[3] A. M. Turing, "On computable numbers, with an application to the entscheidungs problem," *Proc. London Math. Soc.*, vol. 42, no. 2, pp. 230–265, 1936.

[4] L. Adleman, "Molecular computation of solutions to combinatorial problems," *Science*, vol. 266, pp. 1021–1024, Nov. 11, 1994.

[5] D. Deutsch, "Quantum theory, the church-turing principle and the universal quantum computer," *Proc. Roy. Soc. London A*, vol. 400, pp. 97–117, 1985.

[6] R. Karp, "On the computational complexity of combinatorial problems," *Networks*, vol. 5, pp. 45–68, 1975.

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

[8] S. Imre and F. Balazs, *Quantum Computation and Communications: An Engineering Approach*. Hoboken, NJ, USA: Wiley, 2007.