Quantum speedup in solving the maximal-clique problem

Weng-Long Chang,[1,*] Qi Yu,[2] Zhaokai Li,[2,3] Jiahui Chen,[2,4] Xinhua Peng,[2,3,5,†] and Mang Feng[5,6,7,‡]

[1]*Department of Computer Science, National Kaohsiung University of Applied Sciences, Kaohsiung City 80778, Taiwan, China*

[2]*CAS Key Laboratory of Microscale Magnetic Resonance and Department of Modern Physics, University of Science and Technology of China, Hefei 230026, China*

[3]*Synergetic Innovation Center of Quantum Information and Quantum Physics, University of Science and Technology of China, Hefei 230026, China*

[4]*Institute for Quantum Computing and Department of Physics and Astronomy, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1*

[5]*Synergetic Innovation Center for Quantum Effects and Applications, Hunan Normal University, Changsha 410081, China*

[6]*State Key Laboratory of Magnetic Resonance and Atomic and Molecular Physics, Wuhan Institute of Physics and Mathematics, Chinese Academy of Sciences, Wuhan 430071, China*

[7]*Department of Physics, Zhejiang Normal University, Jinhua 321004, China*

(Received 16 October 2017; published 29 March 2018)

The maximal-clique problem, to find the maximally sized clique in a given graph, is classically an NP-complete computational problem, which has potential applications ranging from electrical engineering, computational chemistry, and bioinformatics to social networks. Here we develop a quantum algorithm to solve the maximal-clique problem for any graph G with n vertices with quadratic speedup over its classical counterparts, where the time and spatial complexities are reduced to, respectively, $O(\sqrt{2^n})$ and $O(n^2)$. With respect to oracle-related quantum algorithms for the NP-complete problems, we identify our algorithm as optimal. To justify the feasibility of the proposed quantum algorithm, we successfully solve a typical clique problem for a graph G with two vertices and one edge by carrying out a nuclear magnetic resonance experiment involving four qubits.

DOI: 10.1103/PhysRevA.97.032344

I. INTRODUCTION

In a social network, identifying the largest group of people with mutual acquaintance, i.e., who all know each other, is an NP-complete problem [1], whose complexity scales exponentially with the number of people involved in the social network. It is mathematically termed the maximal-clique problem [2,3], described by a graph with vertices and edges representing, respectively, the people and their mutual relations. Its solution is to find the largest group (or groups) with the most vertices connected mutually by edges. Besides its applications in social networks, the clique problem has also been applied to electrical engineering for designing efficient circuits [4], computational chemistry for exploring bound chemicals in the database [5], and bioinformatics for studying evolutionary trees of species or predicting protein structure [6].

Mathematically, the maximal-clique problem is defined regarding a graph $G = (V, E)$ with n vertices and θ edges, where V is a finite set of n vertices in G and E is a set of θ edges connecting pairs of vertices in G. A clique is a set of vertices in which all the vertices are connected with each other by edges. As such, the maximal-clique problem is to find the largest clique in the graph, which has been proven to be NP-complete. Figure 1 shows an example of a graph with six vertices for such a problem where the vertices $\{1,2,3,4\}$ form the largest clique. It has been shown that any brute-force solution to the maximal-clique problem requires an exponential increase of time with the size of the problem [i.e., with time complexity of $O(2^n)$] [2,3] and no effective approximation had been found to solve the clique problem [7]. DNA computing techniques, under the condition of exponentially increasing volumes of DNA (spatial complexity), claimed to solve this problem in linearly increasing time due to operations in parallel [8,9]. However, this is not true in real operations since the maximum number of vertices in procession is limited to 27 [8].

Quantum computers promise to exploit the remarkable properties of quantum mechanical systems to solve certain problems more efficiently than their classical counterparts. Besides the celebrated Shor's algorithm for integer factorization [10] and Grover's algorithm for searching an unsorted database [11], some other quantum algorithms for various hard problems have recently been proposed [12–15]. On the other hand, experimental progress has witnessed the successful implementation of various quantum algorithms, such as factoring algorithm in different quantum computer candidates [16–22] and efficient execution of boson sampling [23–26]. However, some of the aforementioned works are not for NP-complete problems. In principle, the NP-complete problems can be solved by the oracle-related search algorithms, such as Grover search. It is already known that any such oracle-based quantum algorithm could not perform better than quadratic speedup over its classical counterparts [27,28]. This implies that an oracle-related quantum algorithm for the maximal-clique problem, if behaving optimally with finite spatial complexity, should work in time $O(\sqrt{2^n})$.

*changwl@cc.kuas.edu.tw
†xhpeng@ustc.edu.cn
‡mangfeng@wipm.ac.cn

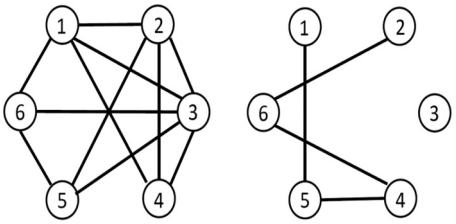

FIG. 1. Typical graph with six vertices and eleven edges (left) and its complementary graph (right). As mentioned in the text, using the complementary graph, we may find the illegal cliques more easily by identifying the vertices connected by edges.

A quantum adiabatic algorithm has been proposed to solve the maximal-clique problem [29]. Unfortunately, asymptotic analysis of quantum adiabatic evolution algorithms appears to be difficult. Here we propose an optimal oracle-related quantum algorithm, based on a quantum circuit oracle model, for solving the maximal-clique problem with quadratic speedup over its classical counterparts. By representing the vertices by qubits, we first filter out the illegal cliques (defined later) under quantum logical gates; then we identify the maximally sized subset of vertices in the legal cliques, followed by operations of a Grover search for the target states representing the solutions of the clique problem. The key point of our algorithm is the polynomial time complexity of the oracle's job for labeling the target states. As a justification of the feasibility of our algorithm, a four-qubit nuclear magnetic resonance (NMR) experiment is accomplished to solve a typical clique problem for a graph G with two vertices and one edge.

II. QUANTUM ALGORITHM

For a graph with n vertices, we require n qubits representing the vertices, and there are 2^n possible cliques from $|0\cdots0\rangle$ to $|1\cdots1\rangle$, where $|0\cdots0\rangle$ and $|1\cdots1\rangle$ represent, respectively, the clique with no vertex and the one with n vertices. The qubit state $|1\rangle$ ($|0\rangle$) represents the presence (absence) of the corresponding vertex in the clique. For the example in Fig. 1, the maximally sized set {1,2,3,4} can be represented as $|111100\rangle$. An efficient way to solve a clique problem of the graph is to consider its complementary graph $\bar{G} = (V, \bar{E})$ with the edges of the vertices out of the set E. For convenience of description, we introduce the definitions of legal clique and illegal clique. A legal clique is the one with no edge in $\bar{G}$ and thus other cliques in $\bar{G}$ belong to illegal cliques (i.e., the vertices connected by edges in $\bar{G}$ form illegal cliques). After excluding the illegal cliques by means of $\bar{G}$, we are able to find the solution from the legal cliques by identifying the maximally sized subset of vertices in G. For example, the state $|1x_2x_3x_41x_6\rangle$, with $x_i = 1$ or 0 for the graph in Fig. 1, evidently denotes an illegal clique. After removing the illegal cliques, we then explore the maximally sized set from the legal cliques as the solution and set it as a target state. The final step is to find the target state by iterating the Grover search operations.

A. Steps of the solution

The concrete implementation of the quantum algorithm is as follows:

(1) Preparing a uniform superposition state. We first prepare a uniform superposition state of n qubits $\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle$ involving all 2^n possible cliques by individually performing a Hadamard gate on each qubit initially prepared in $|0\rangle$. We call this register the data register.

(2) Excluding the illegal cliques. For the graph G with n vertices and θ edges, it is easy to find its complementary graph $\bar{G}$ with n vertices and $m = n(n-1)/2 - \theta$ edges. Any two vertices v_i and v_j disconnecting in the original graph G are connected in the complementary graph $\bar{G}$, i.e., the edges in $\bar{G}$ are represented by $\bar{e}_k = (v_i, v_j)$, where $1 \leqslant k \leqslant m$. Therefore, we remove the sets represented by $|x_1x_2\cdots1_i\cdots1_j\cdots x_{n-1}x_n\rangle$ from all possible cliques. If the kth edge exists in $\bar{G}$, the formula $\overline{x_i \wedge x_j}$ is then of the true value. As such, the requested condition for deciding a legal clique among 2^n possible cliques is that the formula $\wedge_{k=1}^{m}(\overline{x_i \wedge x_j})$ is true. To accomplish the logical flowchart in Fig. 2(a) in a quantum computer, we have to introduce some auxiliary qubits such as $|\bar{e}_k\rangle$ $(1 \leqslant k \leqslant m)$, $|c_0\rangle$, and $|c_k\rangle$ $(1 \leqslant k \leqslant m)$. The operations of $x_i \wedge x_j$ and $\overline{x_i \wedge x_j}$ can be realized by a Toffoli gate when the target bit is initially set to 0 and 1, respectively, i.e., $|x_i\rangle|x_j\rangle|0\rangle \rightarrow |x_i\rangle|x_j\rangle|x_i \wedge x_j\rangle$ and $|x_i\rangle|x_j\rangle|1\rangle \rightarrow |x_i\rangle|x_j\rangle|\overline{x_i \wedge x_j}\rangle$. Consequently, all $|\bar{e}_k\rangle$ and $|c_0\rangle$ are initially set to be $|1\rangle$, while the $|c_k\rangle$ $(1 \leqslant k \leqslant m)$ are initialized as $|0\rangle$. The quantum circuit is shown in Fig. 2(b). Only if the final value of c_m is 1 do we obtain the legal cliques.

(3) Classifying the legal cliques. In order to find the largest cliques in the legal cliques, we first classify the cliques $|x_1, \ldots, x_n\rangle$ into different registers by their Hamming weights (i.e., the number of 1's that appear in the binary representation $x_1\cdots x_n$) as sketched in Fig. 2(c). This idea can be described by the logic flowchart in Fig. 2(d). Auxiliary qubits $z_{i+1,j+1}$ and $z_{i+1,j}$ are employed to store the results of the formulas $x_{i+1} \wedge z_{i,j}$ and $\overline{x_{i+1}} \wedge z_{i,j}$, respectively, which are likely implemented by Toffoli gates. For $0 \leqslant i \leqslant n$ and $0 \leqslant j \leqslant i$, $|z_{i+1,j}\rangle$ and $|z_{i+1,j+1}\rangle$ are all initially prepared in state $|0\rangle$. After the loops are completed, we have successfully classified the legal cliques into $n+1$ qubits $z_{n,i}$, with $i = 0, \ldots, n$, according to their Hamming weight from 0 to n. If at least $|z_{n,i}(x)\rangle = |1\rangle$, the legal cliques with the Hamming weight i exist; otherwise, if for all 2^n, $|z_{n,i}(x)\rangle = |0\rangle$, there are no legal cliques with the Hamming weight i.

(4) Identifying the legal cliques. After the classification of the legal cliques by their Hamming weights, we can identify the largest ones which correspond to the biggest Hamming weight among $n+1$ registers by a quantum Grover search algorithm. That is, the largest cliques are labeled by $|z_{n,i_{\max}}(x)\rangle = |1\rangle$ and $i_{\max}$ is the size of the largest cliques in the graph G. The Grover algorithm is repeatedly applied to the data register x and the qubit $z_{n,i}$, where i starts from n and stops at $i_{\max}$. For the Grover algorithm, an oracle qubit O initialized in $|1\rangle$ is introduced to perform the oracle operation, i.e., to invert those target cliques x labeled by $|z_{n,i}(x)\rangle = |1\rangle$, which is completed by a Hadamard gate on the oracle qubit O and a CNOT gate between qubits $z_{n,i}$ and O. This functions as $|x\rangle|z_{n,i}(x)\rangle|1\rangle \rightarrow |x\rangle|z_{n,i}(x)\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}} \rightarrow (-1)^{z_{n,i}(x)}|x\rangle|z_{n,i}(x)\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}}$. Thus, by applying the reversal quantum circuit of steps 2 and 3, one can accomplish the oracle operation on the data register: $\sum_{x\in\{\{\text{illegal cliques}\}\cap\{\text{legal cliques}|_{z_{n,i}(x)=0}\}\}}|x\rangle - \sum_{x\in\{\text{legal cliques}|_{z_{n,i}(x)=1}\}}|x\rangle$. Then the oracle operation is followed

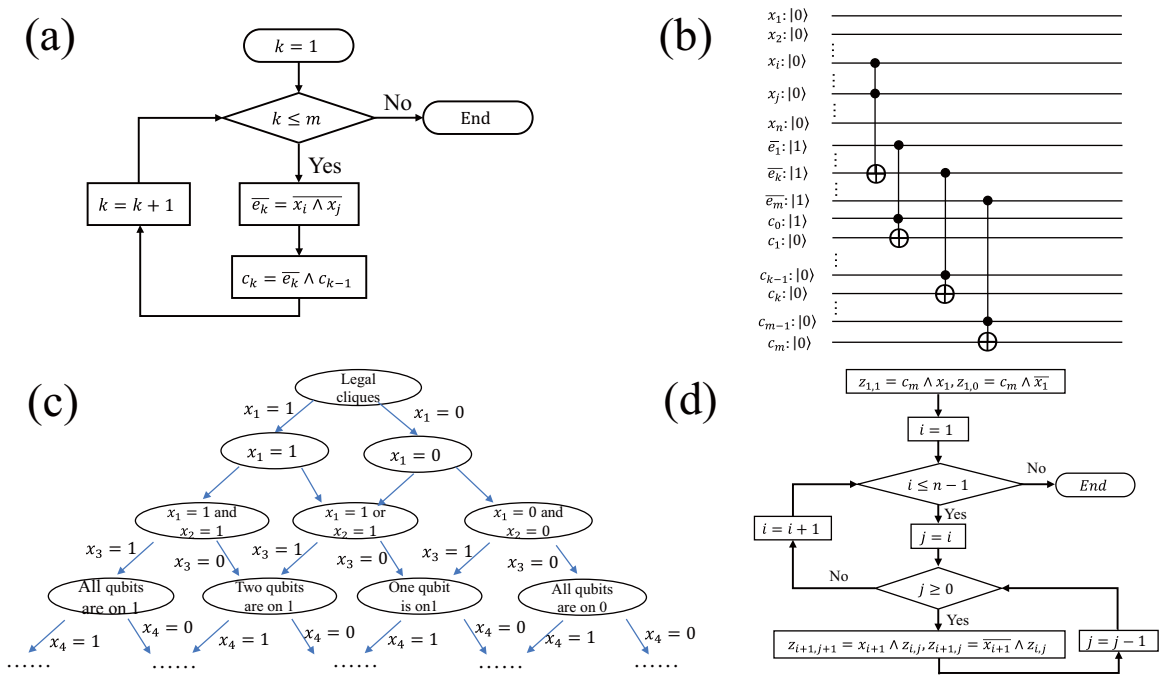

FIG. 2. Quantum mechanical treatment of the maximal-clique problem of a graph. (a) Logical flowchart of deciding a legal clique. The formula $\overline{x_i} \wedge \overline{x_j}$ is realized by the Toffoli gate when the target qubit is in $|1\rangle$ and the formula $\bar{e}_k \wedge c_{k-1}$ is achieved by the Toffoli gate when the target qubit is in $|0\rangle$. Following the steps, we continue the loop by increasing k. (b) Quantum circuit corresponding to (a). Here x_k ($k = 1,2,\ldots$) represent states of qubits encoding the kth vertex v_k, $\bar{e}_k$ ($k = 0,1,2,\ldots$) are states of auxiliary qubits for storing states of the illegal and legal cliques, which are all initialized as $|1\rangle$, and c_k ($k = 0,1,2,\ldots$) are states of auxiliary qubits for further assistance as explained in the flowchart. Here c_0 is initialized as $|1\rangle$ and the rest are initially set to be $|0\rangle$. (c) Tree diagram for classifying the legal cliques according to the Hamming weight of x. In the diagram, we first divide the legal cliques into two registers conditional on the value of x_1. Then we divide the two registers into three depending on the value of x_2. After n steps, we can classify all the legal cliques into $n+1$ registers. These registers contain the cliques with different numbers of vertices from 0 to n. (d) Logical flowchart for counting the number of the vertices. The formulas $x_{i+1} \wedge z_{i,j}$ and $\overline{x_{i+1}} \wedge z_{i,j}$ are carried out cyclically to store their results in the auxiliary qubits $z_{i+1,j+1}$ and $z_{i+1,j}$, respectively. Implementation of $c_m \wedge x_1$ and $c_m \wedge \overline{x_1}$ in the auxiliary qubits $z_{1,1}$ and $z_{1,0}$ is the beginning of the loop.

by Hadamard transforms and a conditional phase shift on the data register to complete one iteration of Grover algorithm [11]. According to the results of the Grover algorithm [11], one requires $O(\sqrt{2^n/M})$ Grover iterations in order to obtain a solution to the search problem with high probability. Here M is the number of solutions. When M is unknown, a quantum counting algorithm [30] can be used to estimate the number M of solutions.

B. Example

To further clarify the algorithm introduced above, we present in Fig. 3(a) a simple example for a graph $G = (3,2)$ with $V = \{v_1, v_2, v_3\}$ and $E = \{e_1 = (v_1, v_2), e_2 = (v_2, v_3)\}$, whose complementary graph is $\bar{G} = (3,1)$ with $V = \{v_1, v_2, v_3\}$ and $\bar{E} = \{\bar{e}_1 = (v_1, v_3)\}$. In this case, three qubits, as the data register, are initialized in the equal superposition state $\frac{1}{2\sqrt{2}} \sum_{x=0}^{7} |x\rangle$. According to $\bar{G}$, a Toffoli gate among v_1, v_3, and $\bar{e}_1$ transforms the initial state $\frac{1}{2\sqrt{2}} \sum_{x=0}^{7} |x\rangle|\bar{e}_1 = 1\rangle$ into $\frac{1}{2\sqrt{2}}(\sum_{x\in\text{illegal}} |x\rangle|\bar{e}_1 = 0\rangle + \sum_{x\in\text{legal}} |x\rangle|\bar{e}_1 = 1\rangle)$. Thus the illegal cliques $x = x_1x_2x_3 \in \{(x_1 = 1), (x_3 = 1)\}$ are separated from the legal ones by the state of the qubit $\bar{e}_1$, i.e., the legal cliques are labeled by $|\bar{e}_1 = 1\rangle$ in the second term. Here the register $|c_0\rangle \cdots |c_m\rangle$ can be omitted because of only one edge in $\bar{G}$. Let $|z\rangle = |z_{1,0}z_{1,1}, \ldots, z_{3,0}z_{3,1}, \ldots, z_{3,3}\rangle$. After the classification by the Hamming weight, the output state is $\frac{1}{2\sqrt{2}}[\sum_{x\in\text{illegal}} |x\rangle|\bar{e}_1 = 0\rangle|z = 0\rangle + \sum_{x\in\text{legal}} |x\rangle|\bar{e}_1 = 1\rangle|z(x)\rangle]$, where the values of $z(x)$ are listed in Table I. Thus the legal clique x with the Hamming weight i is classified into the qubit $z_{3,i}(x)$, i.e., $z_{3,1}$ is used to store the legal cliques with only one of x_1, x_2, or x_3 being in 1. Here $z_{3,1}(x) = 1$ implies that x exists in the legal cliques and so on.

Then the final task is to find the maximal value $i_{\max}$ with $z_{3,i_{\max}}(x) = 1$ and the corresponding x. The Grover search algorithm is employed to complete this task. Starting from $z_{3,3}$, due to all $z_{3,3}(x) = 0$, Grover iterations fail to return a valid solution, illustrating no legal cliques with the Hamming weight 3. Then we execute the algorithm by restarting and in the final step the Grover algorithm is moved to $z_{3,2}$. With around four applications of the Grover iteration, we may finally obtain two valid solutions $|110\rangle$ and $|011\rangle$, implying that the largest cliques are $\{v_1, v_2\}$ and $\{v_2, v_3\}$. The algorithm ends. The quantum circuit for the example is shown in Figs. 3(b) and 3(c).

C. Estimate of complexity

It is not difficult to access the complexity of this quantum algorithm. Given a graph with n vertices, the required number

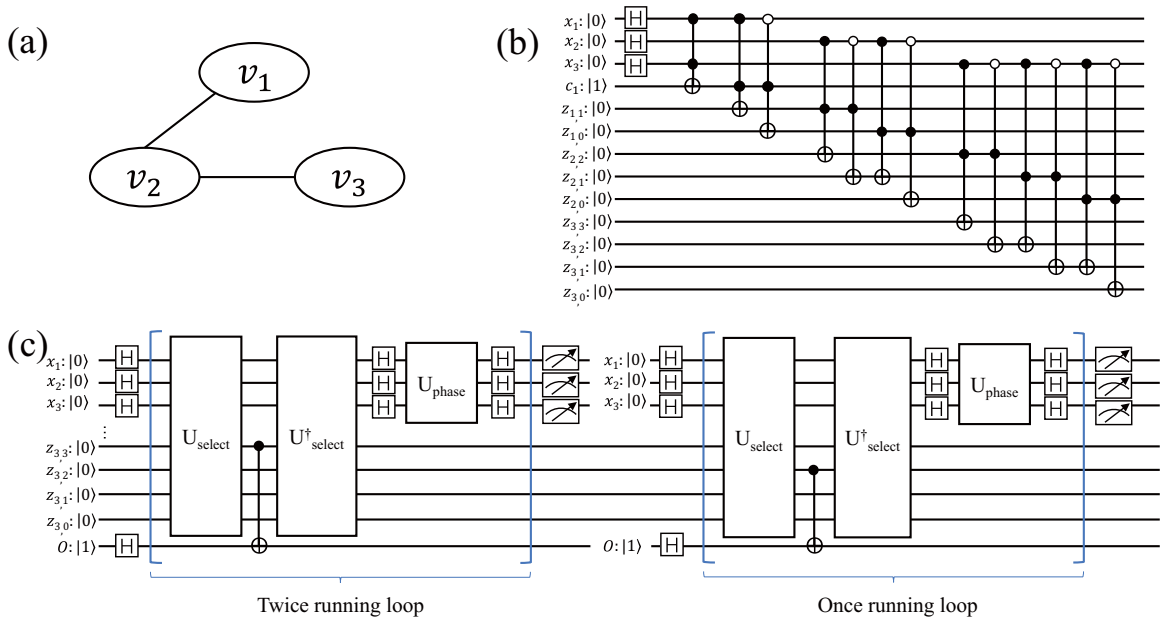

FIG. 3. (a) Graph $G(3,2)$ with $V = \{v_1, v_2, v_3\}$ and $E = \{e_1 = (v_1, v_2), e_2 = (v_2, v_3)\}$. (b) Quantum circuit to perform the algorithm. (c) Diagram of quantum circuit to complete the algorithm. Here U_{select} denotes the execution of the quantum circuit in (b). The oracle $|O\rangle$ initialized in $|1\rangle$ is introduced for Grover iterations which include the oracle's operation, Hadamard gates, and a conditional phase shift on the data register, as explained in the text. Grover iterations are repeatedly applied to qubits from $z_{3,3}$ to $z_{3,0}$ and stop when we find a valid solution.

of qubits, including those considered as auxiliary, is at most $2m + n + 2 + n(n+3)/2$, implying a spatial complexity $O(n^2)$, a polynomial increment with n. However, the time complexity of the Grover search algorithm is $O(\sqrt{2^n})$. This implies that, in addition to the polynomial time increase of the oracle's job for identifying the target states, the time complexity of our algorithm is at most $O(n^3\sqrt{2^n}) \sim O(\sqrt{2^n})$ in the worst case. Therefore, the maximal-clique problem can be solved with a quadratic speedup by our quantum algorithm in comparison with that by the classical counterparts. Some detailed discussion can be found in Appendix A.

III. EXPERIMENTAL IMPLEMENTATION

To verify our proposed algorithm, we have also accomplished a proof-of-principle NMR experiment for the simplest clique problem for a graph $G = (2,1)$ that consists of two vertices $\{v_2, v_1\}$ and an edge $\{(v_1, v_2)\}$. After optimizing the quantum circuit, as detailed in Appendix B, we only require four qubits to solve this problem, as shown in Fig. 4(a). The four qubits $|x_1\rangle$, $|x_2\rangle$, $|z_{1,1}\rangle$, and $|O\rangle$ are initially prepared as $|0\rangle$, $|0\rangle$, $|0\rangle$, and $|1\rangle$, respectively. The experiment is carried out on a Bruker AV-400 NMR spectrometer (9.4 T) at 303.0 K with the sample iodotrifluoroethylene dissolved in chloroform-d, where three ^{19}F nuclei and a ^{13}C nucleus constitute a four-qubit quantum processor. The natural Hamiltonian of this four-spin quantum system in the double-rotating frame is given by [31,32]

$$H_{\text{NMR}} = \sum_{j=1}^{4} \pi \nu_j \sigma_z^j + \sum_{1 \leqslant j < k \leqslant 4} \frac{\pi}{2} J_{jk} \sigma_z^j \sigma_z^k, \qquad (1)$$

where the measured parameters are shown in Fig. 4(b). The chemical shifts ν_j and the J-coupling constants J_{jk} are, respectively, listed in the diagonal and off-diagonal terms.

We first prepare a pseudopure state ρ_{0000} using the line-selective pulse method [33] with the fidelity of 97.23%. Here the fidelity is calculated by $F = \sqrt{\langle 0000|\rho_{0000}|0000\rangle}$. Then we use a high-fidelity shaped pulse found by the gradient ascent pulse engineering (GRAPE) algorithm [34] to realize the quantum circuit in Fig. 4(a). The GRAPE pulse has a duration of 26 ms, with a theoretical fidelity above 99.85%. The last step is to measure the output state encoding the solution of the clique problem, which only requires the occupation information on the computational basis states from $|0000\rangle$ to $|1111\rangle$. To reconstruct the populations, we record the experimental spectrum for each spin after a $\pi/2$ readout pulse to this spin. Since the natural abundance of ^{13}C in the sample is about 1%, we read out all four spins via the ^{13}C channel, by applying SWAP gates between each ^{19}F spin and the ^{13}C spin, to distinguish those molecules against the large background [35]. The experimental spectra are shown in Fig. 4(c), where the intensities of the main resonant lines are, respectively,

TABLE I. Values of $z(x)$ in the output state after the quantum circuit in Fig. 3(b) is performed.

x_1	x_2	x_3	$\bar{e}_1$	$z_{1,1}$	$z_{1,0}$	$z_{2,2}$	$z_{2,1}$	$z_{2,0}$	$z_{3,3}$	$z_{3,2}$	$z_{3,1}$	$z_{3,0}$
0	0	0	1	0	1	0	0	1	0	0	0	1
0	0	1	1	0	1	0	0	1	0	0	1	0
0	1	0	1	0	1	0	1	0	0	0	1	0
0	1	1	1	0	1	0	1	0	0	1	0	0
1	0	0	1	1	0	0	1	0	0	0	1	0
1	0	1	0	0	0	0	0	0	0	0	0	0
1	1	0	1	1	0	1	0	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0

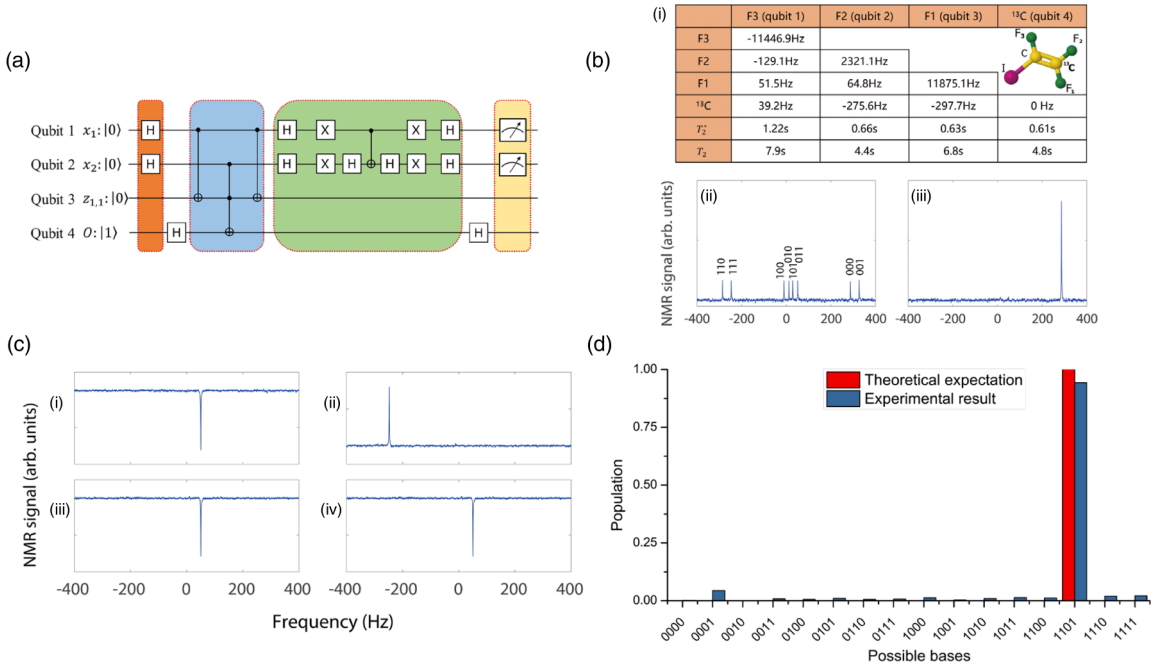

FIG. 4. The NMR experiment for a maximal-clique problem of the graph $G(2,1)$. (a) Simplified quantum circuit for solving the clique problem of the graph $G(2,1)$. (b) Relevant parameters of (i) the iodotrifluoroethylene molecule along with the experimental ^{13}C spectra of (ii) the thermal equilibrium state and (iii) the pseudopure state after a $\pi/2$ pulse on ^{13}C. Eight resonance lines of ^{13}C are labeled by the corresponding states of the three ^{19}F spins. (c) Experimental spectra after performing the quantum circuit in (a) by a $\pi/2$ readout pulse to spin i. The spectra of ^{19}F spins were recorded via the ^{13}C channel, by applying SWAP gates for (i) ^{13}C, (ii) F_1, (iii) F_2, and (iv) F_3. (d) Reconstructed populations of the states from $|0000\rangle$ to $|1111\rangle$ from the experimental results in comparison with the theoretical expectations. The final state is evolved close to $|1101\rangle$ encoding the solution of the clique problem $|x_1x_2\rangle = |11\rangle$, i.e., $\{v_1, v_2\}$.

$-0.9085, 0.9122, -0.9381$, and -0.9672, in comparison to the pseudopure state. From these, the reconstructed populations [36] on the 16 computational basis states are plotted in Fig. 4(d), where the population on $|1101\rangle$ is around 0.9429, much larger than those of other states in Fig. 4(d). Therefore, this experimental result implies that we found the solution of the clique problem of the graph $G(2,1)$: $|x_1x_2\rangle = |11\rangle$, i.e., $\{v_1, v_2\}$, with a high probability 94.29%.

The experimental errors are mainly caused from the imperfect initial-state preparation (∼2.8%), the GRAPE pulse error (∼1.0%), and the imperfect readout pulses (∼2.0%). Decoherence during the implementation is negligible due to the fact that the experimental running time is less than 30 ms, much shorter than the shortest relaxation time 600 ms.

IV. CONCLUSION

We have proposed an optimal oracle-related quantum algorithm based on a quantum circuit oracle model to solve the maximal-clique problem for any graph G with n vertices and θ edges. Our NMR experimental performance, although only solving a simple clique problem, gives us hopes that if our quantum algorithm works efficiently, we would have reason to eagerly await a quantum computer capable of running larger numbers of qubits to practically treating the graph-relevant NP-complete problems with quadratic speedup.

An open question regarding the extension of our quantum algorithm is whether other NP-complete problems can also be optimally solved if they are reduced to the clique problem. Our current answer is negative since solving a certain NP-complete problem in this way would probably take a time complexity more than $O(2^n)$ (see Appendix C for details), which makes the quantum treatment less efficient than the classical treatment [with time complexity of $O(2^n)$]. Further clarification of this point is needed.

ACKNOWLEDGMENTS

This work was supported by National Key Basic Research Program of China (Grants No. 2013CB921800, No. 2014CB848700, and No. 2017YFA0304503), National Science Fund for Distinguished Young Scholars of China (Grant No. 11425523), National Natural Science Foundation of China (Grants No. 11674360, No. 11375167, and No. 11227901), the Strategic Priority Research Program (B) of the CAS (Grants No. XDB01030400 and No. XDB21010100), and Key Research Program of Frontier Sciences of the CAS (Grant No. QYZDY-SSW-SLH004).

APPENDIX A: COMPLEXITY ASSESSMENT

To solve the maximal-clique problem of a graph G with n vertices and θ edges (the complementary graph $\bar{G}$ thus has n vertices and $m = \frac{n(n-1)}{2} - \theta$ edges), we require n Hadamard gates to generate the uniform superposition state, $2m$ Toffoli gates to exclude illegal cliques, and $n(n+1)$ NOT gates and $n(n+1)$ Toffoli gates to classify legal cliques by their Hamming weights. This indicates that we also need $n(n+1)$ NOT gates and $2m + n(n+1)$ Toffoli gates to restore the qubits

to their initial states. The steps described above, including a CNOT and a Hadamard gate, are employed to complete the oracle's work of the Grover search. Then the Grover search is performed. The Grover operator can be decomposed into $H^{\otimes n}U_{\mathrm{PSG}}H^{\otimes n}$, where U_{PSG} is an n-qubit conditional phase-shift gate defined as

$$U_{\mathrm{PSG}}:\begin{cases}|x\rangle \rightarrow -|x\rangle & \text{for } x \neq 0\\ |0\rangle \rightarrow |0\rangle & \text{otherwise.}\end{cases} \tag{A1}$$

Thus one requires $O(\sqrt{2^n/M})$ Grover iterations in order to obtain a solution to the search problem with high probability. Here M is the number of solutions. Therefore, the numbers of logic gates required to solve the clique problem are

$$\begin{aligned}
&O(2n2^{n/2}+n+1)\sim O(n2^{n/2}) && \text{(Hadamard gate)},\\
&O(2(n^2+n)2^{n/2})\sim O(n^2 2^{n/2}) && \text{(NOT gate)},\\
&O(2^{n/2}) && \text{(CNOT gate)},\\
&O([4m+2(n^2+n)]2^{n/2})\sim O((m+n^2)2^{n/2}) && \text{(Toffoli gate)},\\
&O(2^{n/2}) && \text{(conditional phase-shift gate of } n \text{ qubits)},\\
&O(1) && \text{(measurement)}.
\end{aligned} \tag{A2}$$

For the worst case, we need to repeat the algorithm n times. Thus the complexity is

$$\begin{aligned}
&O(n(2n2^{n/2}+n+1))\sim O(n^2 2^{n/2}) && \text{(Hadamard gate)},\\
&O(2n(n^2+n)2^{n/2})\sim O(n^3 2^{n/2}) && \text{(NOT gate)},\\
&O(n2^{n/2}) && \text{(CNOT gate)},\\
&O(n[4m+2(n^2+n)]2^{n/2})\sim O(n(m+n^2)2^{n/2}) && \text{(Toffoli gate)},\\
&O(n2^{n/2}) && \text{(conditional phase-shift gate of } n \text{ qubits)},\\
&O(n) && \text{(measurement)}.
\end{aligned} \tag{A3}$$

As such, the average gate complexity is equal to $(1+2+\cdots+n)/n$ multiplied by the complexity of the best case,

$$\begin{aligned}
&O\left(\frac{n+1}{2}(2n2^{n/2}+n+1)\right)\sim O(n^2 2^{n/2}) && \text{(Hadamard gate)},\\
&O((n+1)(n^2+n)2^{n/2})\sim O(n^3 2^{n/2}) && \text{(NOT gate)},\\
&O\left(\frac{n+1}{2}2^{n/2}\right)\sim O(n2^{n/2}) && \text{(CNOT gate)},\\
&O((n+1)[2m+(n^2+n)]2^{n/2})\sim O(n(m+n^2)2^{n/2}) && \text{(Toffoli gate)},\\
&O\left(\frac{n+1}{2}2^{n/2}\right)\sim O(n2^{n/2}) && \text{(conditional phase-shift gate of } n \text{ qubits)},\\
&O\left(\frac{n+1}{2}\right)\sim O(n) && \text{(measurement)}.
\end{aligned} \tag{A4}$$

In experiments the Hadamard gates for initializing the system can be implemented in parallel, which means that the time complexity associated with Hadamard gates for initialization should actually be divided by n. Nevertheless, this does not affect the asymptotic complexity since its dominant part is due to the Hadamard gates used in the Grover operator.

The spatial complexity can be obtained by counting the number of qubits required in the algorithm. In summary, one needs n data qubits to encode the graph with n vertices, $2m+1$ auxiliary qubits ($\bar{e}$ and c quantum registers) to exclude illegal cliques, and $\frac{n(n+3)}{2}$ qubits (z quantum register) to classify legal cliques. In addition, the qubit $|O\rangle$ is needed for the Grover search. In total, $O(2m+n+2+\frac{n(n+3)}{2})\sim O(n^2+m)$ qubits are required. The numbers of qubits needed in both the best and the worst cases are the same since qubits can be reused. Note that $\sim$ means asymptotically equivalent [$g(n)$ is said to be asymptotically equivalent to $f(n)$ if $\lim_{n\rightarrow\infty} g(n)/f(n) < \infty$].

Therefore, we consider that our proposed oracle-related quantum algorithm for the maximal-clique problem behaves with polynomial-scaled spatial complexity and $O(\sqrt{2^n})$-scaled time complexity. Based on the fact that any oracle-related quantum algorithm cannot work better than quadratic speedup over its classical counterparts [27,28], we identify our proposed quantum algorithm to be optimal with respect to currently known quantum algorithms.

APPENDIX B: QUANTUM CIRCUIT FOR OUR EXPERIMENT

This appendix explains how to reach a four-qubit quantum circuit, as plotted in Fig. 4(a), for a clique problem regarding the graph G with two vertices and one edge.

All the possible cliques in G are $\{v_1,v_2\}$, $\{v_2\}$, $\{v_1\}$, and $\varnothing$, where the maximum-sized clique is $\{v_1,v_2\}$. The quantum

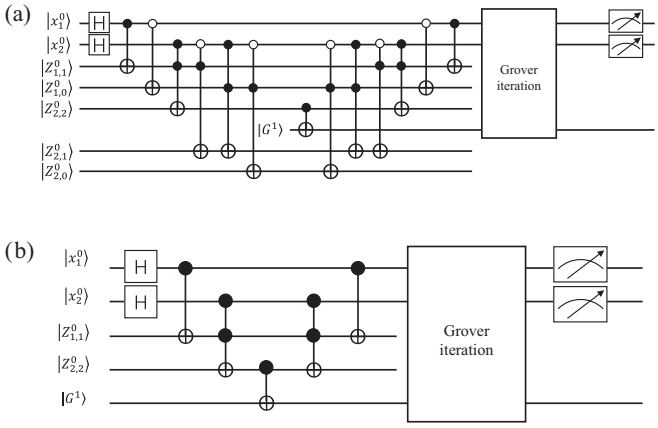

FIG. 5. (a) Standard version of the quantum circuit for solving the clique problem and (b) simplified quantum circuit for solving the clique problem. The superscript in $|\cdot\rangle$ represents the initial state of the qubit in $|0\rangle$ or $|1\rangle$.

circuit based on the standard steps as described in the main text should involve seven qubits [see Fig. 5(a)]. However, since there is no clique in the complementary graph of G, we can skip the step of excluding illegal cliques and simplify the circuit. According to the flowchart in Fig. 2(a), we describe the algorithm in Fig. 5(b), where the qubits $|z_{2,1}^0\rangle$ and $|z_{2,0}^0\rangle$ are excluded from the circuit, because we can obtain the answer in the first Grover iteration and measurement. Besides, the qubit $|z_{1,0}^0\rangle$ works as a control on the qubits $|z_{2,1}^0\rangle$ and $|z_{2,0}^0\rangle$. Since we have deleted $|z_{2,1}^0\rangle$ and $|z_{2,0}^0\rangle$, the qubit $|z_{1,0}^0\rangle$ can also be discarded.

In fact, the quantum circuit can be further simplified. Since the qubit $|z_{2,2}^0\rangle$ is designed only to control the oracle qubit $|O\rangle$, this qubit $|z_{2,2}^0\rangle$ can also be reduced if we carefully consider the initial state of $|O\rangle$. Therefore, we finally get to the four-qubit quantum circuit, as in Fig. 4(a), for solving the clique problem regarding the graph G.

APPENDIX C: EXTENSION OF OUR QUANTUM ALGORITHM

We argue below that it is impossible to optimally solve other NP-complete problems by simply reducing them to the clique problem under consideration.

For simplicity, we exemplify the three-satisfiability (3-SAT) problem with n Boolean variables and m clauses, in which each clause contains three Boolean variables. The solution to 3-SAT is to find whether there is a truth assignment that satisfies all the clauses. In terms of the approaches in [37–39], such a 3-SAT problem can be reduced to a clique problem with $2n + 3m$ vertices and $[(2n + 3m)(2n + 3m - 1)/2 - (n + 6m)]$ edges. However, solving such a clique problem, with our optimal oracle-related quantum algorithm, will take a time complexity $O(\sqrt{2^{2n+3m}}) > O(2^n)$. This implies that, by this way of reduction, the quantum solution of the 3-SAT problem is less efficient than a classical treatment.

Further exploration is needed to clarify this problem. If this way of reduction is in principle unavailable for a quantum treatment, developing independent quantum algorithms for different NP-complete problems will be indispensable.

[1] R. D. Luce and A. D. Perry, Psychometrika **14**, 95 (1949).
[2] R. Karp, Networks **5**, 45 (1975).
[3] M. R. Garey and D. S. Johnson, *Computer and Intractability: A Guide to the Theory of NP-completeness* (Freeman, San Francisco, 1979).
[4] M. C. Paull and S. H. Unger, IRE Trans. Electron. Comput. **ECC8**, 356 (1959).
[5] N. Rhodes, P. Willett, A. Calvet, J. B. Dunbar, and C. Humblet, J. Chem. Inf. Comput. Sci. **43**, 443 (2003).
[6] R. Samudrala and J. Moult, J. Mol. Biol. **279**, 287 (1998).
[7] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy, in *Proceedings of the 32nd IEEE Symposium on Foundations of Computer Science* (IEEE, Piscataway, 1991), p. 2.
[8] Q. Ouyang, P. D. Kaplan, S. Liu, and A. Libchaber, Science **278**, 446 (1997).
[9] M. Ho, W.-L. Chang, M. Guo, and L. T. Yang, IEICE Trans. Inf. Syst. **E-87D**, 1782 (2004).
[10] P. Shor, in *Proceedings of the 35th Annual Symposium on the Foundations of Computer Science*, edited by S. Goldwasser (IEEE Computer Society, Los Alamitos, 1994), p. 124.
[11] L. K. Grover, Phys. Rev. Lett. **79**, 325 (1997); **79**, 4709 (1997); **80**, 4329 (1998).
[12] A. W. Harrow, A. Hassidim, and S. Lloyd, Phys. Rev. Lett. **103**, 150502 (2009).
[13] S. Aaronson and A. Arkhipov, in *Proceedings of the ACM Symposium on Theory of Computing* (ACM, New York, 2011), p. 333.
[14] S. P. Jordan, K. S. M. Lee, and J. Preskill, Science **336**, 1130 (2012).
[15] J. Li, X. Peng, J. Du, and D. Suter, Sci. Rep. **2**, 260 (2012).
[16] L. M. K. Vandersypen, M. Steffen, G Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, Nature (London) **414**, 883 (2001).
[17] C.-Y. Lu, D. E. Browne, T. Yang, and J.-W. Pan, Phys. Rev. Lett. **99**, 250504 (2007).
[18] B. P. Lanyon, T. J. Weinhold, N. K. Langford, M. Barbieri, D. F. V. James, A. Gilchrist, and A. G. White, Phys. Rev. Lett. **99**, 250505 (2007).
[19] X. Peng and D. Suter, Europhys. Lett. **84**, 40006 (2008).
[20] X. Peng, Z. Liao, N. Xu, G. Qin, X. Zhou, D. Suter, and J. Du, Phys. Rev. Lett. **101**, 220405 (2008); N. Xu, J. Zhu, D. Lu, X. Zhou, X. Peng, and J. Du, *ibid.* **108**, 130501 (2012); Z. Li, N. S. Dattani, X. Chen, X. Liu, H. Wang, R. Tanburn, H. Chen, X. Peng, and J. Du, arXiv:1706.08061.
[21] E. Lucero *et al.*, Nat. Phys. **8**, 719 (2012).
[22] T. Monz, D. Nigg, E. A. Martinez, M. F. Brandl, P. Schindler, R. Rines, S. X. Wang, I. L. Chuang, and R. Blatt, Science **351**, 1068 (2016).
[23] J. B. Spring *et al.*, Science **339**, 798 (2013).
[24] M. A. Broome, A. Fedrizzi, S. Rahimi-Keshari, J. Dove, S. Aaronson, T. C. Ralph, and A. G. White, Science **339**, 794 (2013).
[25] M. Tillmann, B. Dakić, R. Heilmann, S. Nolte, A. Szameit, and P. Walther, Nat. Photon. **7**, 540 (2013).

[26] A. Crespi, R. Osellame, R. Ramponi, D. J. Brod, E. F. Galvão, N. Spagnolo, C. Vitelli, E. Maiorino, P. Mataloni, and F. Sciarrino, Nat. Photon. **7**, 545 (2013).

[27] L. K. Grover, in *Proceedings of the 28th Annual ACM Symposium on Theory of Computing* (ACM, New York, 1996), p. 212.

[28] C. H. Bennett, E. Bernstein, G. Brassard, and U. V. Vazirani, SIAM J. Comput. **26**, 1510 (1997).

[29] A. M. Childs, E. Farhi, J. Goldstone, and S. Gutmann, Quantum Inf. Comput. **2**, 181 (2002).

[30] G. Brassard, P. Hoyer and A. Tapp, in *Proceedings of the 25th International Colloquium on Automata, Languages, and Programming*, Lecture Notes in Computer Science Vol. 1443 (Springer, Berlin, 1998), p. 820.

[31] L. M. Vandersypen and I. L. Chuang, Rev. Mod. Phys. **76**, 1037 (2005).

[32] M. H. Levitt, *Spin Dynamics: Basics of Nuclear Magnetic Resonance* (Wiley, New York, 2001).

[33] X. Peng, X. Zhu, X. Fang, M. Feng, K. Gao, X. Yang, and M. Liu, Chem. Phys. Lett. **340**, 509 (2001).

[34] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, J. Magn. Reson. **172**, 296 (2005).

[35] X. Peng, Z. Luo, W. Zheng, S. Kou, D. Suter, and J. Du, Phys. Rev. Lett. **113**, 080404 (2014).

[36] J.-S. Lee, Phys. Lett. A **305**, 349 (2002).

[37] S. Cook, in *Proceedings of Third Annual ACM Symposium on Theory of Computing* (ACM, New York, 1971), pp. 151–158.

[38] M. R. Garey and D. S. Johnson, *Computer and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, New York, 1979).

[39] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms* (MIT Press, Cambridge, 2001).