

## A Quantum Algorithm for Finding a Hamilton Circuit\*

GUO Hao,<sup>1</sup> LONG Gui-Lu,<sup>1,2,3,4,5,†</sup> SUN Yang<sup>1,4,6,7</sup> and XIU Xiao-Lin<sup>8</sup>

<sup>1</sup>Department of Physics, Tsinghua University, Beijing 100084, China

<sup>2</sup>Institute of Theoretical Physics, The Chinese Academy of Sciences, Beijing 100080, China

<sup>3</sup>Centre for Nuclear Theory, Lanzhou National Laboratory of Heavy Ions, The Chinese Academy of Sciences, Lanzhou 730000, China

<sup>4</sup>The Key Laboratory of Quantum Information and Measurements, MOE, Beijing 100084, China

<sup>5</sup>Center of Atomic Molecular and Nanosciences, Tsinghua University, Beijing 100084, China

<sup>6</sup>Department of Physics and Astronomy, University of Tennessee, Knoxville, Tennessee 37996, USA

<sup>7</sup>Department of Physics, Xuzhou Normal University, Xuzhou 221009, Jiangsu Province, China

<sup>8</sup>Department of Electronic Engineering, Beijing Institute of Technology, Beijing 100081, China

(Received November 18, 2000; Revised March 14, 2001)

**Abstract** A quantum algorithm for solving the classical NP-complete problem — the Hamilton circuit is presented. The algorithm employs the quantum SAT and the quantum search algorithms. The algorithm is square-root faster than classical algorithm, and becomes exponentially faster than classical algorithm if nonlinear quantum mechanical computer is used.

**PACS numbers:** 03.67.Lx, 89.70.+c

**Key words:** quantum algorithm, Hamilton circuit, NP-problem

### 1 Introduction

NP and NP-complete problems are one of the important themes of research in quantum computing. It is well known that all NP-complete problems are equivalent in classical computation theory. Whether there exists an algorithm to solve NP-complete problems in polynomial time is an essential question. In 1971, Cook proved that all other NP problems can be converted to the “satisfiability” problem (Usually it is abbreviated to SAT problem) in polynomial time. So, if the SAT problem can be solved in polynomial time then all other NP problems can also be. But in classical computational theory no polynomial algorithm has been found for the SAT problem yet. Recently, a quantum algorithm is given by Masanori and Masuda<sup>[1]</sup> to solve it in polynomial time under the assumption that any superposition of orthogonal vectors in space  $C^{2^n}$  can be physically detected efficiently. However this assumption is difficult to be realized in practice.

In this paper, we present a quantum algorithm for solving the famous NP-complete problem that whether there exist Hamilton circuits in a specific graph by using a part of the method that solves the SAT problem and Grover’s searching algorithm. Our algorithm can solve the problem in time bounded by  $\log n * \sqrt{n}$  on any input of length  $n$ , and this is much faster than classical algorithm. And

it can even be finished in polynomial time if a nonlinear quantum “searching” machine put forward by Abrams and Lloyd<sup>[2]</sup> is employed.

### 2 Notation and Analysis

The SAT problem plays an important role in computational complexity research. Before we state the SAT problem, we first introduce some basic notions.

Assume that  $X = \{x_1, x_2, \dots, x_n\}$  be a Boolean set. Then  $x_k$  and its negation  $\bar{x}_k$  ( $k = 1, 2, \dots, n$ ) are called literals and the set of all such literals are denoted by

$$X' = \{x_1, \bar{x}_1, x_2, \bar{x}_2, \dots, x_n, \bar{x}_n\}.$$

A subset of elements of  $X'$  can form a clause, for example  $C' = \{x_1, \bar{x}_2, x_3\}$ . The truth value of clause  $C'$ ,  $t(C')$ , is defined as

$$t(C') \equiv \bigvee_{x \in C'} t(x),$$

that is, if one or more of the literals in a clause is true, the truth value of the clause is true, and it is false if all the literals are false. These clauses can form a set of clauses  $O = \{C_1, C_2, \dots, C_m\}$ . The truth value of set  $O$  is defined as

$$t(O) = \bigwedge_{i=1}^m t(C_i),$$

that is, the truth value of set  $O$  is true if every clause in the set is true, and otherwise it is false. We say a clause

\*The project supported in part by National Natural Science Foundation of China, the Fok Ying Tung Education Foundation, Major State Basic Research Development Program under contract No. G200077400 and the HangTian Science Foundation

<sup>†</sup>Corresponding author, mailing address: Department of Physics, Tsinghua University, Beijing 100084, China

$C'$  is satisfiable if and only if  $t(C') = 1$ . Similarly, satisfiability is defined for  $O$  as  $t(O) = 1$ , that is to say  $O$  is satisfiable if and only if all clauses in  $O$  is satisfiable.

The SAT problem is given below.

Given a set

$$X = \{x_1, \dots, x_n\}$$

and a set

$$O = \{C_1, C_2, \dots, C_m\}$$

of clauses, the problem is whether there exists a truth assignment of all the  $x$ 's to make  $O$  satisfiable. To evaluate the truth value of  $O$  for a given assignment of the literals is easy and it can be carried out in polynomial time. However it is difficult to check if a Boolean set  $O$  is satisfiable or not. It generally involves by exhaustive searching: one inputs every combination of the assignments of literals into the expression to check if  $O$  is satisfiable. Since they are  $2n$  literals, the number of trials is the order of  $2^{2n}$ .

In graph theory, one is interested in problems such as if a graph has a certain property. A linear graph, or simply called a graph hereafter,  $G = (V, E)$  consists of a set of vertices  $V = \{v_1, v_2, \dots\}$ , and another edge set

$E = \{e_1, e_2, \dots\}$ . Each edge  $e_k$  is identified with an unordered pair  $(v_i, v_j)$  of vertices if the vertices  $v_i$  and  $v_j$  are both incident on the edge  $e_k$ . We can also use  $(e_i, e_j)$  to identify a vertex at which the pair of edges intersects. A graph  $G$  is said to be connected if there is at least one path between every pair of vertices. A Hamilton circuit in a connected graph is defined as a closed path that traverses every vertex of  $G$  exactly once, except of course the starting vertex, at which the path also terminates. Hence a Hamilton circuit in a graph of  $n$  vertices consists of exactly  $n$  edges.

The Hamilton circuit problem is to find the Hamilton circuits for a given graph. It is easy to check if a given path is a Hamilton circuit or not. However, it is difficult to check if a given graph has Hamilton circuits and to find them. According to the definition of a Hamilton circuit, the problem can be phrased as: given a graph  $G = \{V, E\}$  with  $n$  vertices, does there exist a permutation  $f$  of the vertices, such that the rearranged vertex set

$$V' = \{v_{f(1)}, v_{f(2)}, \dots, v_{f(n)}\}$$

gives an edge set  $E'$  which is formed by connecting every pair of successive vertices,

$$E' = \{(v_{f(1)}, v_{f(2)}), (v_{f(2)}, v_{f(3)}), \dots, (v_{f(n-1)}, v_{f(n)}), (v_{f(n)}, v_{f(1)})\},$$

which is just the edge set  $E$  or a subset of  $E$  of the graph? Here  $f(1), f(2), \dots, f(n)$  is a permutation of the  $n$  vertices,  $(v_{f(1)}, v_{f(2)})$  indicates the edge from vertex  $v_{f(1)}$  to vertex  $v_{f(2)}$ . Apparently, the edge formed by the rearranged vertex set  $V'$  is a Hamilton path if all the elements of its edge set  $E'$  belong to the edge set of the graph. If some of the edges formed by the rearranged vertices do not belong to the edge set of the graph, the circuit formed by these vertices is not a valid path of the graph, hence not a circuit of the graph.

To solve the Hamilton circuit problem, we need the properties of a Hamilton circuit. In a given graph  $G = (V, E)$ , a subset  $g$  of the edge set  $E$  is said to cover graph  $G$  if every vertex in  $G$  is incident on at least one edge in  $g$ , in other words, every vertex in  $G$  is connected to at least one edge in  $g$ . We say  $g$  is the edge covering set of  $G$ .

If  $G$  has a Hamilton circuit  $H$ , let  $E'$  be the set of the edges of  $H$ , then according to the definition of a Hamilton circuit,  $E'$  must be the edge covering set of  $G$ . It is easy to verify that  $H$  has the following properties:

- (i)  $|E'| = |V|$ , namely the number of the edges of  $H$  is equal to the number of the vertices of graph  $G$ .
- (ii) Each vertex is incident on just two edges of  $E'$ .

Conversely, we can see that an edge covering set with the properties mentioned above must be a Hamilton circuit of  $G$ . Thus if  $G$  has no edge covering sets, it has no Hamilton circuits. The algorithm that we present in the next section uses these properties.

### 3 Quantum Algorithm of Finding Hamilton Circuits

Let the vectors

$$|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad |1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

in the Hilbert space  $C^2$  denote the truth values 0 and 1 of the Boolean lattice  $L$  respectively, so an element  $x \in X$  can be denoted by  $|0\rangle$  and  $|1\rangle$ . In order to describe a clause  $C$  with at most  $n$  elements, we need the  $n$ -tuple Hilbert space  $H = \otimes_1^n C^2$ . In fact, the truth value of a set  $O$  of clauses can be looked upon as a Boolean function  $f(x)$  on the set  $X$  of  $n$  Boolean variables. The essential part of the SAT algorithm given in Ref. [2] is to construct a quantum network to exploit the function  $f$  by using the fundamental quantum gates such as Not gate and Controlled-Not gates. Beginning with a quantum computer and preparing it in a uniform weighted superposition of all possible inputs. Then performing the unitary transformation

corresponding function  $f$  on the inputs, and then we get a superposed state of all possible output because of the quantum parallelism, which is the well-known advantage of quantum computation. Having obtained all the possible outputs, Ohya and Masuda algorithm assumes the distinguishability of 0 and 1 to get the final results.

In our Hamilton circuit algorithm, we also use quantum network to implement the clauses. After this, one needs to find whether there exists an input value  $x$  for which  $f(x) = 1$ . We can use the nonlinear quantum “searching” algorithm<sup>[2-4]</sup> which finds the marked state in polynomial steps, or the Grover algorithm<sup>[5]</sup> which achieves only quadratic speedup over classical algorithms. Nonlinear quantum computer involves nonlinear quantum mechanics, and its realization is even more difficult than the linear quantum computers that are being extensively studied.

Now we give the method to solve Hamilton circuit problem. Let  $G = (V, E)$  be a given graph, and  $m = |E|$  is the number of edges in the edge set  $E$ . We define  $m$  in “edge” Boolean variables  $x_1, x_2, \dots, x_m$  corresponding to the  $m$  edges of  $G$ . In other words, the edge set of  $G$ , which is denoted as  $E$ , can be written as  $E = \{x_1, x_2, \dots, x_m\}$ . For a vertex  $v_i \in E$ , we define a “vertex” clause  $C_i = \{x_{i1}, x_{i2}, \dots, x_{ik}\}$ , in which Boolean variables  $x_{i1}, x_{i2}, \dots, x_{ik}$  are the  $k$  edges that connect to vertex  $v_i$ .

Now we will find an edge covering set of  $G$ . Denoting  $n = |V|$ , the number of elements in  $V$ . Let  $D$  denote a subset of  $E$ ,  $D = \{x_{k1}, x_{k2}, \dots\}$ . We assume the truth value  $t(x_i) = 1$  if and only if  $x_i \in D$ , that is to say, giving a specific subset  $D$  has actually given the truth assignment of all edge Boolean variables in  $E$ . Then the truth value of a clause  $C_i$  is

$$t(C_i) = \vee_{x \in C_i} t(x).$$

If  $t(C_i) = 1$ , at least one  $x_i$  is nonzero in clause  $C_i$ , or to say that vertex  $v_i$  is incident on at least one edge contained in  $D$ . It is easy to see that  $D$  is an edge covering set of  $G$  if

$$\bigwedge_{j=1}^n \vee_{x \in C_j} t(x) = 1.$$

Let  $X = \{C_1, C_2, \dots, C_n\}$ , we call  $X$  a set of “vertex” clause. Thus the problem to find an edge covering set  $D$  of  $G$  is equivalent to finding a truth assignment of  $E$  which makes

$$\bigwedge_{j=1}^n \vee_{x \in C_j} t(x) = 1.$$

The algorithm to find a Hamilton circuit of graph  $G$  is given below.

**Step 1** For a specific graph  $G$ , we use the “edge” Boolean variables and “vertex” clauses to find an edge

covering set  $D$  of  $G$ , using the first part of the algorithm of SAT problem given in Ref. [1]. In this step we need to construct a quantum network to implement the functions. The number of steps needed for this purpose is polynomial in  $m$  and  $n$ , and we denote this by  $p(m, n)$ . Masanori and Masuda gave the exact expression of  $p(m, n)$  which is  $11mn - 3m$ , where  $m$  and  $n$  denote the number of  $E$  and  $V$  respectively.

**Step 2** In this step we must check the output from step 1 and determine which input makes the output equal 1. By finding the cases which give 1 in the output, we get the edge covering set of the graph. As has been discussed before, we can use Grover’s quantum search algorithm<sup>[5]</sup> or its generalizations. This step needs  $\sqrt{2^m}$  steps. In nonlinear quantum computer, this can be further accelerated.

**Step 3** Verify if the set  $D$  satisfies the two properties of Hamilton circuit which we gave in Sec. 2. If it satisfies the properties, then  $D$  is a Hamilton circuit of  $G$ , otherwise it is not. Obviously this step can be completed in polynomial steps.

The total number of steps is the order of  $(11mn - 3m)\sqrt{2^m}$  if we use Grover’s algorithm in searching the solution, and can be polynomial if nonlinear quantum searching is used.

It is worth while discussing the Grover algorithm. In Grover’s original algorithm, there is only one marked state. Here the number of marked states may be greater than 1, and the generalized algorithm that applies to cases with more than 1 marked state.<sup>[6]</sup> If the dimension of problem is not too big, we need also to use a generalization that replaces the phase inversions with arbitrary phase rotations.<sup>[7-10]</sup> If we use nonlinear quantum searching algorithm given in Ref. [2] instead of Grover’s searching in step 2, we may finish the work in polynomial steps. In fact nonlinear quantum search can be thought of as an extension of Grover’s database search algorithm.

It is stressed that without nonlinear quantum mechanics, present quantum computers cannot speed up unsorted database search exponentially. Zalka has shown<sup>[11]</sup> that Grover algorithm is optimal. In other words, square-root speed-up is the best one in searching a marked state in an unsorted database. Shor<sup>[12]</sup> algorithm can factorize a large number in polynomial steps. However, it still remains unresolved if the factorization of a large number is NP complete or not. The exponential speed-up is achieved in the simulation of a quantum system.<sup>[13]</sup> It is still an

open question that if quantum computer can solve NP-complete problems other than the simulation of quantum systems in polynomial steps.

To summarize, we have given an quantum algorithm for solving the Hamilton's circuit problem by using the SAT quantum algorithm and a quantum searching algorithm. If the Grover's quantum search algorithm is used,

the algorithm solves the problem in  $O(\sqrt{n})$  where  $n$  is the size of the problem. If a nonlinear quantum searching algorithm is used, the Hamilton's algorithm becomes polynomial (It will be an interesting problem to study the structure of nonlinear quantum searching in terms of nonlinear algebra<sup>[14]</sup>). If linear quantum computer is used, quadratic speed-up will be achieved.

---

## References

- [1] Masanori Ohya and Natsuki Masuda, *Open Systems and Information Dynamics* **7** (2000) 33.
- [2] D. Abrams and S. Lloyd, *Phys. Rev. Lett.* **81** (1998) 3992.
- [3] Marek Czachor, *quant-ph/9802051*, v2, 23 Feb. (1998).
- [4] Steven Weinberg, *Ann. Phys.* **194** (1989) 336.
- [5] L.K. Grover, *Phys. Rev. Lett.* **79** (1997) 325.
- [6] L.K. Grover, *Phys. Rev. Lett.* **7** (1997) 325.
- [7] G.L. LONG, *et al.*, *Commun. Theor. Phys.* (Beijing, China) **32** (1999) 335.
- [8] G.L. LONG *et al.*, *Phys. Lett.* **A262** (1999) 27.
- [9] G.L. LONG *et al.*, *J. Phys.* **A34** (2001) 867.
- [10] G.L. LONG *et al.*, *Phys. Lett.* **A61** (2000) 042305.
- [11] C. Zalka, *Phys. Rev.* **A60** (1999) 2746.
- [12] P. Shor, *Proceedings of the 35th Annual Symposium on Foundations of Computer Science, 1994*, IEEE Computer Society Press, Los Alamos, CA (1994) p. 124.
- [13] R. Feynman, *Int. J. Theor. Phys.* **21** (1982) 467.
- [14] D. RUAN and W. RUAN, *Phys. Lett.* **263** (2000) 78.