NEW QUANTUM ALGORITHM FOR STUDYING NP-COMPLETE PROBLEMS

MASANORI OHYA

Tokyo University of Science, Department of Information Sciences,
Noda City, Chiba 278-8510, Japan
(e-mail: ohya@is.noda.tus.ac.jp)

and

IGOR V. VOLOVICH

Steklov Mathematical Institute, Gubkin St. 8, 117966 Moscow, Russia
(e-mail: volovich@mi.ras.ru)

(*Received December 6, 2002*)

Ordinary approach to quantum algorithm is based on quantum Turing machine or quantum circuits. It is known that this approach is not powerful enough to solve NP-complete problems. In this paper we study a new approach to quantum algorithm which is a combination of the ordinary quantum algorithm with a chaotic dynamical system. We consider the satisfiability problem as an example of NP-complete problems and argue that the problem, in principle, can be solved in polynomial time by using our new quantum algorithm.

Keywords: Quantum algorithm, NP-complete problem, chaotic dynamics.

1. Introduction

Ordinary approach to quantum algorithm is based on quantum Turing machine or quantum circuits [1–3]. It is known that this approach is not powerful enough to solve NP-complete problems [4, 5]. In [6] we have proposed a new approach to quantum algorithm which goes beyond the standard quantum computation paradigm. This new approach is a sort of combination of the ordinary quantum algorithm and a chaotic dynamics. This approach was based on the results obtained in the paper [7].

There are important problems such as the knapsack problem, the travelling salesman problem, the integer programming problem, the subgraph isomorphism problem, the satisfiability problem that have been studied for decades and for which all known algorithms have a running time that is exponential in the length of the input. These five and many other problems belong to the set of NP-complete problems [4].

Many NP-complete problems have been identified, and it seems that such problems are very difficult and probably exponential. If so, solutions are still needed,

and in this paper we consider an approach to these problems based on quantum computers and chaotic dynamics as mentioned above.

As in the previous papers [7, 6], we again consider the satisfiability problem as an example of NP-complete problems and argue that the problem, in principle, can be solved in polynomial time by using our new quantum algorithm.

It is widely believed that quantum computers are more efficient than classical computers. In particular, Shor [8, 9] gave a remarkable quantum polynomial-time algorithm for the factoring problem. However, it is known that this problem is not NP-complete but is NP-intermediate.

Since the quantum algorithm of the satisfiability problem (SAT for short) has been considered in [7], Accardi and Sabbadini showed that this algorithm is combinatoric one and they discussed its combinatoric representation [10]. It was shown in [7] that the SAT problem can be solved in polynomial time by using a quantum computer under the assumption that a special superposition of two orthogonal vectors can be physically detected. The problem one has to overcome here is that the output of computations could be a very small number and one needs to amplify it to a reasonable large quantity.

In this paper we construct a new model (representation) of computations which combine ordinary quantum algorithm with a chaotic dynamical system and prove that one can solve the SAT problem in polynomial time.

For a recent discussion of computational complexity in quantum computing see [11–14]. Mathematical features of quantum computing and quantum information theory are summarized in [15].

2. SAT Problem

Let $X \equiv \{x_1, \ldots, x_n\}$ be a set. Then x_k and its negation $\overline{x}_k$ $(k = 1, 2, \ldots, n)$ are called *literals* and the set of all such literals is denoted by $X' = \{x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n\}$. The set of all subsets of X' is denoted by $\mathcal{F}(X')$ and an element $C \in \mathcal{F}(X')$ is called a *clause*. We take a truth assignment to all variables x_k. If we can assign the truth value to at least one element of C, then C is called *satisfiable*. When C is satisfiable, the truth value $t(C)$ of C is regarded as true, otherwise, that of C is false. Take the truth values as true "1", false "0". Then

$$C \text{ is satisfiable iff } t(C) = 1.$$

Let $L = \{0, 1\}$ be a Boolean lattice with usual join $\vee$ and meet $\wedge$, and let $t(x)$ be the truth value of a literal x in X. Then the truth value of a clause C is written as

$$t(C) \equiv \vee_{x \in C} t(x).$$

Further, the set $\mathcal{C}$ of all clauses C_j $(j = 1, 2, \ldots m)$ is called satisfiable iff the meet of all truth values of C_j is 1,

$$t(\mathcal{C}) \equiv \wedge_{j=1}^{m} t(C_j) = 1.$$

Thus the SAT problem is defined as follows.

DEFINITION 1. *SAT Problem*: Given a set $X \equiv \{x_1, \ldots, x_n\}$ and a set $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$ of clauses, determine whether $\mathcal{C}$ is satisfiable or not.

That is, this problem is to ask whether there exists a truth assignment to make $\mathcal{C}$ satisfiable.

It is known [4] for usual algorithm that the time to check the satisfiability is polynomial only when a specific truth assignment is given, but we cannot determine the satisfiability in polynomial time when an assignment is not specified.

Note that a formula made by the product (AND $\wedge$) of the disjunction (OR $\vee$) of literals is said to be in the *product of sums* (POS) form. For example, the formula

$$(x_1 \vee \bar{x}_2) \wedge (\bar{x}_1) \wedge (x_2 \vee \bar{x}_3)$$

is in the POS form. Thus a formula in the POS form is said to be *satisfiable* if there is an assignment of values to variables so that the formula has value 1. Therefore, the SAT problem can be regarded as determining *whether or not a formula in the POS form is satisfiable.*

The following analytical formulation of the SAT problem is useful. We define a family of Boolean polynomials $f_{\mathcal{A}}$, indexed by the following data. Let $\mathcal{A}$ be a set

$$\mathcal{A} = \{S_1, \ldots, S_N, T_1, \ldots, T_N\},$$

where $S_i, T_i \subseteq \{1, \ldots, n\}$, and $f_{\mathcal{A}}$ be defined as

$$f_{\mathcal{A}}(x_1, \ldots, x_n) = \prod_{i=1}^{N} \left(1 + \prod_{a \in S_i} (1 - x_a) \prod_{b \in T_i} x_b\right).$$

We assume here the addition modulo 2. The SAT problem now is to determine whether or not there exists a value of $\mathbf{x} = (x_1, \ldots, x_n)$ such that $f_{\mathcal{A}}(\mathbf{x}) = 1$.

3. Quantum algorithm

Although the quantum algorithm of the SAT problem is needed to add the dust bits to the input n bits, the number of dust bits is of the order of n [7, 10]. Therefore for simplicity we will work in this paper in the $(n+1)$-tuple tensor product Hilbert space $\mathcal{H} \equiv \otimes_1^{n+1} \mathbb{C}^2$ with the computational basis

$$|x_1, \ldots, x_n, y\rangle = \otimes_{i=1}^{n} |x_i\rangle \otimes |y\rangle,$$

where $x_1, \ldots, x_n$, $y = 0$ or 1. We denote $|x_1, \ldots, x_n, y\rangle = |\mathbf{x}, y\rangle$. The quantum version of the function $f(\mathbf{x}) := f_{\mathcal{A}}(\mathbf{x})$ is given by the unitary operator $U_f |\mathbf{x}, y\rangle = |\mathbf{x}, y + f(\mathbf{x})\rangle$. We assume that the unitary matrix U_f can be build in the polynomial time, see [7]. Now let us use the usual quantum algorithm:

(i) using the Fourier transform produce from $|\mathbf{0},0\rangle$ the superposition

$$|v\rangle := \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}} |\mathbf{x}, 0\rangle ,$$

(ii) use the unitary matrix U_f to calculate $f(\mathbf{x})$,

$$|v_f\rangle = U_f |v\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}} |\mathbf{x}, f(\mathbf{x})\rangle .$$

Now if we measure the last qubit, i.e., apply the projector $P = I \otimes |1\rangle\langle 1|$ to the state $|v_f\rangle$, then we obtain that the probability to find the result $f(\mathbf{x}) = 1$ is $\|P|v_f\rangle\|^2 = r/2^n$, where r is the number of roots of the equation $f(\mathbf{x}) = 1$. If r is suitably large to detect it, then the SAT problem is solved in polynomial time. However, for small r, the probability is very small and this means that in fact we do not get any information about the existence of the solution of the equation $f(\mathbf{x}) = 1$, so that in such a case we need further discussion.

Let us simplify our notation. After the step (ii), the quantum computer will be in the state

$$|v_f\rangle = \sqrt{1-q^2}\,|\varphi_0\rangle \otimes |0\rangle + q\,|\varphi_1\rangle \otimes |1\rangle ,$$

where $|\varphi_1\rangle$ and $|\varphi_0\rangle$ are normalized n qubit states and $q = \sqrt{r/2^n}$. Effectively our problem is reduced to the following 1-qubit problem. We have the state

$$|\psi\rangle = \sqrt{1-q^2}\,|0\rangle + q\,|1\rangle$$

and we want to distinguish between the cases $q = 0$ and $q > 0$ (a small positive number).

It is argued in [5] that quantum computer can speed up NP problems quadratically but not exponentially. The no-go theorem states that if the inner product of two quantum states is close to 1, then the probability that a measurement distinguishes which one of the two occurs is exponentially small. And one could claim that amplification of this distinguishability is not possible.

At this point we emphasize that we do not propose to make a measurement (not read) which will be overwhelmingly likely to fail. What we do is a proposal to use the output $|\psi\rangle$ of the quantum computer as an input for another device which uses chaotic dynamics in the sequel.

The amplification would not be possible if we used the standard model of quantum computations with a unitary evolution. However, the idea of our paper is different. We propose to combine quantum computer with a chaotic dynamics amplifier. Such a quantum chaos computer is a new model of computations going beyond usual scheme of quantum computation and we demonstrate that the amplification is possible in the polynomial time.

One could object that we do not suggest a practical realization of the new model of computations. But at the moment nobody knows how to make a practically useful

implementation of the standard model of quantum computing. Quantum circuit or quantum Turing machine is a mathematical model, though a convincing one. It seems to us that the quantum chaos computer considered in this paper deserves investigation and has a potential to be realizable.

In this paper we propose a mathematical model of computations for solving SAT problem by refining our previous paper [6]. A possible specific physical implementation of quantum chaos computations with some error correction will be discussed in a separate paper [16], which is somehow related to the recently proposed atomic quantum computer [17].

4. Chaotic dynamics

Various aspects of classical and quantum chaos have been the subject of numerous studies, see [18] and references therein. The investigation of quantum chaos by using quantum computers has been proposed in [19–21]. Here we will argue that chaos can play a constructive role in computations.

Chaotic behaviour in a classical system usually is considered as an exponential sensitivity to initial conditions. It is this sensitivity we would like to use to distinguish between the cases $q = 0$ and $q > 0$ from the previous section.

Consider the so-called logistic map which is given by the equation

$$x_{n+1} = ax_n(1 - x_n) \equiv g(x), \qquad x_n \in [0, 1].$$

The properties of the map depend on the parameter a. If we take, for example, $a = 3.71$, then the Lyapunov exponent is positive, the trajectory is very sensitive to the initial value and one has the chaotic behaviour [18]. It is important to notice that if the initial value $x_0 = 0$, then $x_n = 0$ for all n.

It is known [2] that any classical algorithm can be implemented on quantum computer. Our quantum chaos computer will consist of two blocks. One block will be the ordinary quantum computer performing computations with the output $|\psi\rangle = \sqrt{1 - q^2}\,|0\rangle + q\,|1\rangle$. The second block will be a computer performing computations of the *classical* logistic map. These two blocks should be connected in such a way that the state $|\psi\rangle$ should first be transformed into the density matrix of the form

$$\rho = q^2 P_1 + (1 - q^2)\, P_0,$$

where P_1 and P_0 are projectors to the state vectors $|1\rangle$ and $|0\rangle$. This connection would in fact be nontrivial and actually should be considered as the third block. One has to notice that P_1 and P_0 generate an abelian algebra which can be considered as a classical system. In the second block the density matrix ρ above is interpreted as the initial data ρ_0, and we apply the logistic map as

$$\rho_m = \frac{(I + g^m(\rho_0)\sigma_3)}{2},$$

where I is the identity matrix and σ_3 is the z-component of Pauli matrix on $\mathbb{C}^2$. This expression is different from that of our first paper [6]. To find a proper value m we finally measure the value of σ_3 in the state ρ_m such that

$$M_m \equiv \mathrm{tr}\, \rho_m \sigma_3.$$

After simple computation we obtain

$$\rho_m = \frac{(I + g^m(q^2)\sigma_3)}{2}, \quad \text{and} \quad M_m = g^m(q^2).$$

Thus the question is whether we can find such m in polynomial steps of n satisfying the inequality $M_m \geqq \frac{1}{2}$ for very small but nonzero q^2. Here we have to remark that if one has $q = 0$ then $\rho_0 = P_0$ and we obtain $M_m = 0$ for all m. If $q \neq 0$, the stochastic dynamics leads to the amplification of the small magnitude q in such a way that it can be detected as is explained below. The transition from ρ_0 to ρ_m is nonlinear and can be considered as a classical evolution because our algebra generated by P_0 and P_1 is abelian. The amplification can be done within at most $2n$ steps due to the following propositions. Since $g^m(q^2)$ is x_m of the logistic map $x_{m+1} = g(x_m)$ with $x_0 = q^2$, we use the notation x_m in the logistic map for simplicity.

PROPOSITION 2. *For the logistic map* $x_{n+1} = ax_n(1 - x_n)$ *with* $a \in [0, 4]$ *and* $x_0 \in [0, 1]$, *let* x_0 *be* $\frac{1}{2^n}$ *and the set* J *be* $\{0, 1, 2, \ldots, n, \ldots, 2n\}$. *If* a *is* 3.71, *then there exists an integer* m *in* J *satisfying* $x_m > \frac{1}{2}$.

Proof: Suppose that there does not exist such m in J. Then $x_m \leq \frac{1}{2}$ for any $m \in J$. The inequality $x_m \leq \frac{1}{2}$ implies

$$x_m = 3.71(1 - x_{m-1})x_{m-1} \geq \frac{3.71}{2} x_{m-1}.$$

Thus we have

$$\frac{1}{2} \geq x_m \geq \frac{3.71}{2} x_{m-1} \geq \cdots \geq \left(\frac{3.71}{2}\right)^m x_0 = \left(\frac{3.71}{2}\right)^m \frac{1}{2^n},$$

from which we get

$$2^{n+m-1} \geq (3.71)^m.$$

According to the above inequality, we obtain

$$m \leq \frac{n-1}{\log_2 3.71 - 1}.$$

Since $\log_2 3.71 \doteqdot 1.8912$, we have

$$m \leq \frac{n-1}{\log_2 3.71 - 1} < \frac{5}{4}(n-1),$$

which is definitely less than $2n-1$ and it is contradictory to the statement "$x_m \leq \frac{1}{2}$ for any $m \in J$". Thus there exists m in J satisfying $x_m > \frac{1}{2}$. □

PROPOSITION 3. *Let a and n be the same as in the above proposition. If there exists m_0 in J such that $x_{m_0} > \frac{1}{2}$, then $m_0 > \frac{n-1}{\log_2 3.71}$.*

Proof: Since $0 \leq x_m \leq 1$, we have

$$x_m = 3.71(1 - x_{m-1})x_{m-1} \leq 3.71x_{m-1},$$

which reduces to

$$x_m \leq (3.71)^m x_0.$$

For m_0 in J satisfying $x_{m_0} > \frac{1}{2}$, it holds

$$x_0 \geq \frac{1}{(3.71)^{m_0}} x_{m_0} > \frac{1}{2 \times (3.71)^{m_0}}.$$

It follows from $x_0 = \frac{1}{2^n}$ that

$$\log_2 2 \times (3.71)^{m_0} > n,$$

which implies

$$m_0 > \frac{n-1}{\log_2 3.71}.$$ □

According to these propositions, it is enough to check the value x_m (M_m) around the above m_0 when q is $\frac{1}{2^n}$ for a large n. More generally, when $q=\frac{k}{2^n}$ with some integer k, it is easily checked that the above two propositions hold and the value x_m (M_m) becomes over $\frac{1}{2}$ around the m_0 above.

One can think about various possible implementations of the idea of using chaotic dynamics for computations, which is an open and very interesting problem. For this problem, realization of nonlinear quantum gates will be essential; it will be discussed in [16].

Finally, we show in Fig. 1 how we can easily amplify the small q in several steps.

5. Conclusion

The complexity of the quantum algorithm for the SAT problem has been considered in [7] where it was shown that one can build the unitary matrix U_f in the polynomial time. We have also to consider the number of steps m in the classical algorithm for the logistic map performed on quantum computer. It is the probabilistic part of the construction and one has to compute several times to be able to distinguish the cases $q = 0$ and $q > 0$. Thus we conclude that the quantum chaos algorithm can solve the SAT problem in polynomial time according to the above propositions.

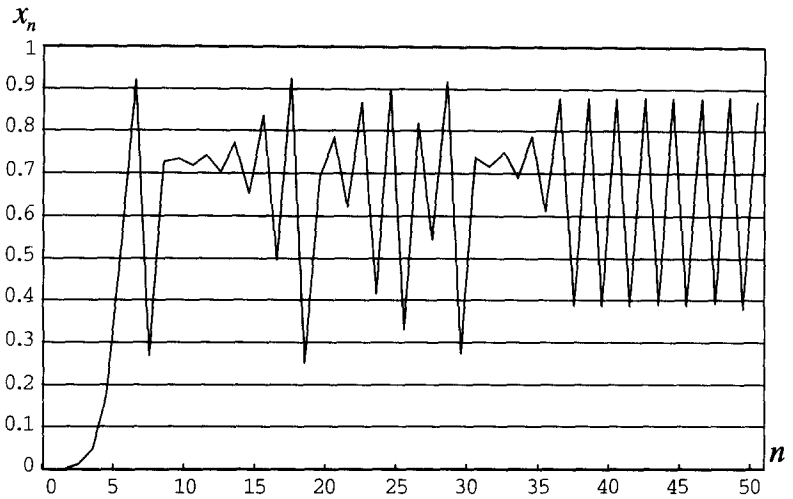

Fig. 1. Change of x_n w.r.t. time n

In conclusion, in this paper the quantum chaos algorithm is proposed. It combines the ordinary quantum algorithm with quantum chaotic dynamics amplifier. We argue that such an algorithm can be powerful enough to solve the NP-complete problems in the polynomial time. Our proposal is to show the existence of algorithm to solve NP-complete problem. The physical implementation of this algorithm is another question and it is strongly desirable to study it further.

REFERENCES

[1] D. Bouwmeester, A. Ekert and A. Zeilinger: *The Physics of Quantum Information*, Springer, Berlin 2001.
[2] D. Deutsch: Quantum theory, the Church–Turing principle and the universal quantum computer, *Proc. Royal Soc. London series A*, **400** (1985), 97–117.
[3] E. Bernstein and U. Vazirani: Quantum Complexity Theory, in *Proc. the 25th Annual ACM Symposium on Theory of Computing*, ACM Press, New York 1993, 11–20.
[4] M. Garey and D. Johnson: *Computers and Intractability—a Guide to the Theory of NP-completeness*, Freeman, 1979.
[5] C. H. Bennett, E. Bernstein, G. Brassard and U. Vazirani: Strengths and Weaknesses of Quantum Computing, quant-ph/9701001
[6] M. Ohya and I. V. Volovich: Quantum computing, NP-complete problems and chaotic dynamics, in T. Hida and K. Saito (eds.), *Quantum Information II*, World Scientific, Singapore 2000; quant-ph/9912100.
[7] M. Ohya and N. Masuda: NP problem in Quantum Algorithm, *Open Systems and Information Dynamics* **7** No.1 (2000), 33–39.
[8] P. W. Shor: Algorithm for quantum computation: Discrete logarithm and factoring algorithm, *Proceedings of the 35th Annual IEEE Symposium on Foundation of Computer Science*, 1994, 124–134.
[9] A. Ekert and R. Jozsa: Quantum computation and Shor's factoring algorithm, *Rev. Mod. Phys.* **68**, No.3 (1996), 733–753.
[10] L. Accardi, R. Sabbadini: On the Ohya–Masuda quantum SAT Algorithm, in *Proceedings International Conference "Unconventional Models of Computations"*, I. Antoniou, C. S. Calude, M. Dinneen (eds.), Springer 2001.

[11] L. Fortnow and J. Rogers: Complexity Limitations on Quantum Computation, cs.CC/9811023.
[12] R. Cleve: An Introduction to Quantum Complexity Theory, quant-ph/9906111.
[13] E. Hemaspaandra, L. A. Hemaspaandra and M. Zimand: Almost-Everywhere Superiority for Quantum Polynomial Time, quant-ph/9910033.
[14] D. S. Abrams and S. Lloyd: Nonlinear quantum mechanics implies polynomial-time solution for NP-complete and #P problems, quant-ph/9801041.
[15] M. Ohya and I. V. Volovich: *Quantum Information, Computation, Cryptography and Teleportation*, Springer (to appear).
[16] M. Ohya and I. V. Volovich: An implementation of chaotic dynamics by atomic computer, in preparation.
[17] I. V. Volovich: Atomic Quantum Computer, quant-ph/9911062.
[18] M. Ohya: Complexities and Their Applications to Characterization of Chaos, *Int. J. Theoret. Phys.* **37** (1998), 495.
[19] S. A. Gardiner, J. I. Cirac and P. Zoller: *Phys. Rev. Lett.* **79** (1997). 4790.
[20] R. Schack, *Phys. Rev. A* **57** (1998), 1634; T. Brun and R. Schack: quant-ph/9807050.
[21] I. Kim and G. Mahler: Quantum Chaos in Quantum Turing Machine, quant-ph/9910068.