数理解析研究所講究録
第 871 巻 1994 年 30-36

量子 Turing 機械による NP 完全問題の多項式時間解法について

On a Method of Solving NP-Complete Problems in Polynomial Time Using the Quantum Turing Machine

三原孝志　　西野哲朗
Takashi Mihara　Tetsuro Nishino

北陸先端科学技術大学院大学 情報科学研究科
School of Information Science
Japan Advanced Institute of Science and Technology, Hokuriku

1 Introduction

Since current computing devices are based on the Turing Machine, principles of computation on those devices are based on classical physics. It is well known that classical physics is sufficient to explain macroscopic phenomena, but is not to explain microscopic phenomena like the interference of electrons. In these days, speed-up and down-sizing of computing devices have been carried out using quantum physical effects, however, principles of computation on these devices are also based on classical physics.

Until now, several physicists have proposed models of computation based on quantum physics, so-called quantum computers. In early models of quantum computers such as those of P. A. Benioff[3], researchers placed a great importance on finding an precise representration of Turing Machines based on quantum physics. Thus, inherent properties of quantum physics were not involved in those models. In other words, their purpose was to design a good simulator of Turing machines based on quantum physics. In 1985, D. Deutsch introduced, for the first time, a model involving a superposition of physical states, which is one of the inherent properties of quantum physics[5], and suggested that his quantum computer might have a potential for a new type of computer. Indeed, Deutsch and Jozsa have shown a problem such that the quantum computer can solve faster than any other classical models of computation by using so-called quantum parallelism[6].

A goal of this paper is to show mathematically a possibility that the Deutsch's universal QTM can compute faster than any other classical models of computation. Namely, in the sequel, we show that the Deutsch's universal quantum Turing machine can solve any NP-complete problem in polynomial time under a physical assumption that we can observe the existence of a specific physical state in a given superposition of physical states.

2 The Quantum Turing Machine

Definition of the Quantum Turing Machine

Like an ordinary Turing machine, a quantum Turing machine $M$ consists of a finite control, an infinite tape, and a tape head.

Definition [4] A *Quantum Turing Machine* (QTM) is a 7-tuple $M = (Q, \Sigma, \Gamma, U, q_0, B, F)$, where

$Q$ is a finite set of *states*,
$\Gamma$ is a *tape alphabet*,
$B \in \Gamma$ is a *blank symbol*,
$\Sigma \subseteq \Gamma$ is an *input alphabet*,
$\delta$ is a *state transition function* and is a mapping from $Q \times \Gamma \times \Gamma \times Q \times \{L, R\}$ to $\mathbf{C}$ (the set of complex numbers),
$q_0 \in Q$ is a *initial state*,
$F \subseteq Q$ is a set of *final states.*

An expression $\delta(p, a, b, q, d) = c$ represents the following: if $M$ reads a symbol $a$ in a state $p$ (let c_1 be this configuration of $M$), $M$ writes a symbol $b$ on the square under the tape head, changes the state into $q$, and moves the head one square in the direction denoted by $d \in \{L, R\}$ (let c_2 be this configuration of $M$), and $c$ is called an *amplitude* of this event. Then we define the probability that $M$ changes its configuration from c_1 to c_2 to be $|c|^2$.

This state transition function $\delta$ defines a linear mapping in a linear space of superpositions of $M$'s configurations. This linear mapping is specified by the following matrix M_δ. Each row and column of M_δ corresponds to a configuration of $M$. Let c_1 and c_2 be two configurations of $M$, then the entry corresponding to c_2 row and c_1 column of M_δ is $\delta$ evaluated at the tuple which transforms c_1 into c_2 in a single step. If no such tuple exists, the corresponding entry is 0. We call this matrix M_δ a *time evolution matrix* of $M$.

> **Assumption**: For any QTM $M$, the time evolution matrix M_δ must be a *unitary matrix.*

Namely, if $M_\delta^\dagger$ is the transpose conjugate of M_δ and $I$ is the unit matrix, then the relations $M_\delta^\dagger M_\delta = M_\delta M_\delta^\dagger = I$ must be satisfied by M_δ.

Computations and Observations

A *computation* by $M$ is an evolution process of a physical system defined by the unitary matrix M_δ. If we denote an initial state of $M$ by $[\psi(0)]$ and a state at time $s$ by $[\psi(s)]$,

$$[\psi(\tau t)] = M_\delta^t[\psi(0)],$$

where $\tau$ is the time required by $M$ to execute a single step.

And, the tape content will be *observed* as follows: if the vector (a superposition of configurations) $\psi = \sum_i \alpha_i c_i$ is written on the tape, for any vector $\phi$, we can observe with probability $|(\phi, \psi)|^2$ that $\psi$ is parallel to $\phi$. Especially, we can observe with probability $|\alpha_i|^2$ that $\psi$ is parallel to c_i.

In this paper, we assume the following on the observation of a configuration.

> **Assumption A:** When we observe a specific configuration $C$, if $C$ exists in a superposition, we can observe the existence of $C$ with certainty.

Actually, the very recent result of Aharonov et al.[1] suggests that this assumption could be a valid one.

The Universal Quantum Turing Machine

Deutsch's universal QTM $U$ can executes all operations of ordinary reversible Turing machines [2], and unitary transformations for 1-bit state space[5]. Notice that the ordinary Turing machines can not execute these unitary transformations. Deutsch's universal QTM can execute the following eight types of transformations:

$$V_0 = \begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix}, \quad V_1 = \begin{pmatrix} \cos\alpha & i\sin\alpha \\ i\sin\alpha & \cos\alpha \end{pmatrix}, \quad V_2 = \begin{pmatrix} e^{i\alpha} & 0 \\ 0 & 1 \end{pmatrix}, \quad V_3 = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix},$$

$$V_4 = V_0^{-1}, \quad V_5 = V_1^{-1}, \quad V_6 = V_2^{-1}, \quad V_7 = V_3^{-1},$$

where $\alpha$ is any irrational multiple of $\pi$. In this paper, we define $\alpha$ as follows:

$$\alpha = \frac{\pi}{4}.$$

A QTM U_0 which simulates a reversible universal DTM $M$ can be constructed in a way that Deutsch showed in [5]. This QTM U_0 runs in the same number of steps as $M$. The universal QTM (UQTM, for short) which we will use in the sequel is the QTM that will be obtained from U_0 by adding the abilities to execute the above eight types of unitary transformations to U_0. Without loss of generality, we use the following convention when we evaluate the execution time of the universal QTM.

> **Convention**: The universal QTM executes a single step of the reversible DTM in a single step, and each one of the eight types of transformations above in a single step.

3 Results

In this section, we show a method of solving SAT in $O(n^2)$ time using Deutsch's UQTM under the assumption A[7]. In order to simulate a reversible universal DTM, the UQTM $U$ has an input tape T_1, a work tape T_2, and a history tape T_3. Let H_1, H_2, and H_3 be the heads on the tapes T_1, T_2, and T_3, respectively (see Figure 1).

As shown in Figure 1, the UQTM $U$ starts the execution given a logical formula $f$ and the number $m$ of variables in $f$ on the work tape T_2, and the program $P$ to simulate on the input tape T_1. On the right-hand side of $P$, infinitely many blank symbols are written. Notice that the length of the description of $P$ is a constant which is independent of the length of an input given on T_2. The program $P$ consists of state transition rules of one-tape standard DTMs and sentences corresponding to the eight unitary transformations above. Let $V(n, i)$ be the sentence corresponding to these unitary transformations, where $0 \leq n \leq 7$. This sentence represents that "apply the transformation V_n on the ith bit of the work tape".

In all configurations in a superposition, $U$ can simulate a single step of $M$ in the same number of steps. The *time complexity* of the UQTM $U$ is the number of steps executed by $U$

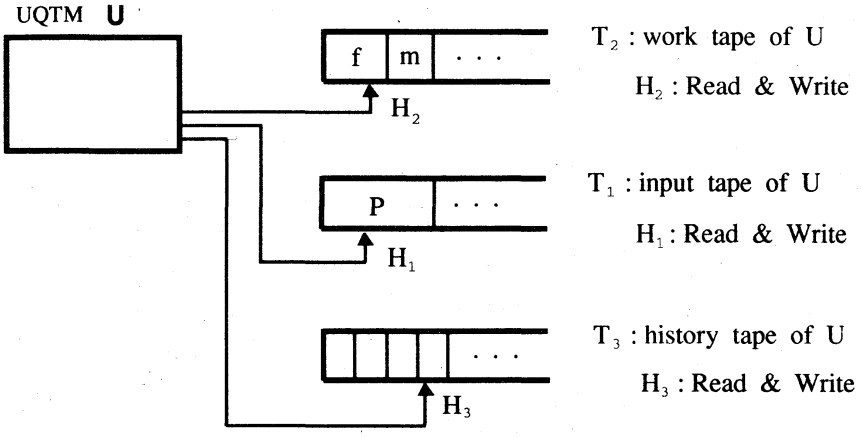

Figure 1: The UQTM U.

until it finally halts, and is represented as a function of the length of the input (in this case, the total length of the descriptions of f and m) given on T_2.

Theorem 3.1 Under the assumption A, the UQTM U can solve SAT in $O(n^2)$ time, where n is the total length of the description of a logical formula f whose satisfiability should be decided and the description of the number of variables in f.

sketch of proof In the sequel, we construct a program P that U simulates in order to solve SAT in $O(n^2)$ time. Let us consider the satisfiability for an m-variable logical formula $f(x_1, x_2, \ldots, x_m)$, where x_i, $i = 1, 2, \ldots, m$ are boolean variables.

The machine U writes an assignments to the variables in the logical formula f together with the corresponding value of f on the tape T_2. We show the program that U will simulate in Figure 2. In the sequel, we explain the behavior of U according to this program.

1. The initial configuration

   The descriptions of the logical formula f and the number m of the variables are given as input on the work tape T_2, and the program P that U simulates is given on the input tape T_1. In the sequel, we identify a configuration of U with a description of an assignment to the variables in f and the corresponding value of f, which are written on the tape T_2 (this description is written on the right-hand side of f and m on T_2). Let $[x_1], \ldots, [x_m]$, and $[x_{m+1}]$ be the bits corresponding to $x_1, \ldots, x_m$ and the value of f under the assignment in this configuration, respectively. In the sequel, we are indicating only $[x_1, \ldots, x_m, x_{m+1}]$ in $[T_2]$ as a configuration of U. Let

$$[\underbrace{0, 0, \ldots, 0}_{m}; 0]$$

   be the initial configuration of U. In order to simplify the presentation, we separate the assignments for the variables from the value of f by semicolon(;).

```
begin
%%% Preparations of inputs
1     for $i = 1$ to $m$ do
2         $V(4, i)$
      od ;
%%% Computation of $f$
3     $x_{m+1} := f(x_1, \ldots, x_m)$ ;
%%% Preparations for an observation
4     for $j = m$ to 1 do
5         $V(0, j)$
      od
end.
```

Figure 2: The program that the UQTM U simulates.

2. Preparations of inputs

There exist 2^m different assignments for m variables of f. Initially, U makes a superposition of configurations corresponding to all of these assignments. U will perform this by applying

$$V_4 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$$

to each bit corresponding to m variables in order. U can execute the transformation by V_4 in a single step. In the execution of the first **for** loop, the initial configuration is transformed as follows:

$$[0, 0, \ldots, 0; 0] \xrightarrow{\mathsf{U}_{x_1} \cdots \mathsf{U}_{x_m}} \frac{1}{\sqrt{2^m}} \sum_{x_1=0}^{1} \sum_{x_2=0}^{1} \cdots \sum_{x_m=0}^{1} [x_1, x_2, \ldots, x_m; 0].$$

Because U can execute each transformation U_{x_i} in a single step, it can execute all the transformations $\mathsf{U}_{x_1}, \cdots, \mathsf{U}_{x_m}$ in m steps.

3. Computation of f

Let U_f be a transformation corresponding to the computation of the value of the logical formula f, then

$$\frac{1}{\sqrt{2^m}} \sum_{x_1=0}^{1} \sum_{x_2=0}^{1} \cdots \sum_{x_m=0}^{1} [x_1, x_2, \ldots, x_m; 0]$$

$$\xrightarrow{U_f} \frac{1}{\sqrt{2^m}} \sum_{x_1=0}^{1} \sum_{x_2=0}^{1} \cdots \sum_{x_m=0}^{1} [x_1, x_2, \ldots, x_m; f(x_1, x_2, \ldots, x_m)] \equiv \psi.$$

In order to execute the third line of the program, U simulates a program of a one-tape standard DTM M_f computing the values of f. U simulates M_f, regarding T_2 as the tape of M_f. Obviously, U can execute the whole computation of above in $O(n^2)$ time.

4. Preparations for an observation

   The machine U executes the reverse transformation of the one executed in the preparations of inputs by applying
   $$V_0 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$$
   to the bits corresponding to the m variables, from $[x_m]$ to $[x_1]$ in order. Because U can execute a transformation by V_0 in a single step, it can execute the **for** loop of the fourth line in Figure 2 in m steps. In this transformation the superposition ψ of U's configurations will be transformed as follows:

   (a) If the values of f for 2^m different assignments are all 0's, ψ will be transformed to one configuration $[0, 0, \ldots, 0; 0]$.

   (b) If the values of f for 2^m different assignments are all 1's, ψ will be transformed to one configuration $[0, 0, \ldots, 0; 1]$.

   (c) Otherwise, ψ will be transformed to a superposition of several configurations.

An Observation

When U completes all the computaions above, we will observe whether there exists a configuration $c_0 = [0, 0, \ldots, 0; 1]$ in the finally obtained superposition of U's configurations. If c_0 exists in the superposition, we conclude that f is satisfiable, and if not, unsatisfiable. The correctness of this decision is shown by the claim below.

> **Claim** f is satisfiable if and only if there exists the configuration $[0, 0, \ldots, 0; 1]$ in the finally obtained superposition of configurations.

The Time Complexity of UQTM *U*

Since the second line in Figure 2 is executed m times in total, U can execute the first **for** loop within $O(m)$ steps. As mentioned above, U can execute the third line within $O(n^2)$ time. Finally, U can also execute the **for** loop of the fourth line within $O(m)$ steps. Therefore, U can execute the procedure in Figure 2 within $O(n^2)$ steps in total. □

Definition [4] A language L is in the class EQP (*exact* or *error-free quantum polynomial time*) if there exists a QTM with a distinguished acceptance tape cell, and a polynomial p, such that given any string x as input, observing the acceptance cell at time $p(n)$ correctly classifies x with respect to L. More generally, a language L is in the class BQP (*bounded-error quantum polynomial time*) if this classification can be accomplished with probability greater than 2/3[4].

Corollary 3.1 Under the assumption A, NP $\subseteq$ EQP.

Example In Figure 3, we show the change of the superpositions of configurations in the case of a function f such that $f(0,0) = f(1,1) = 1, f(0,1) = f(1,0) = 0$.

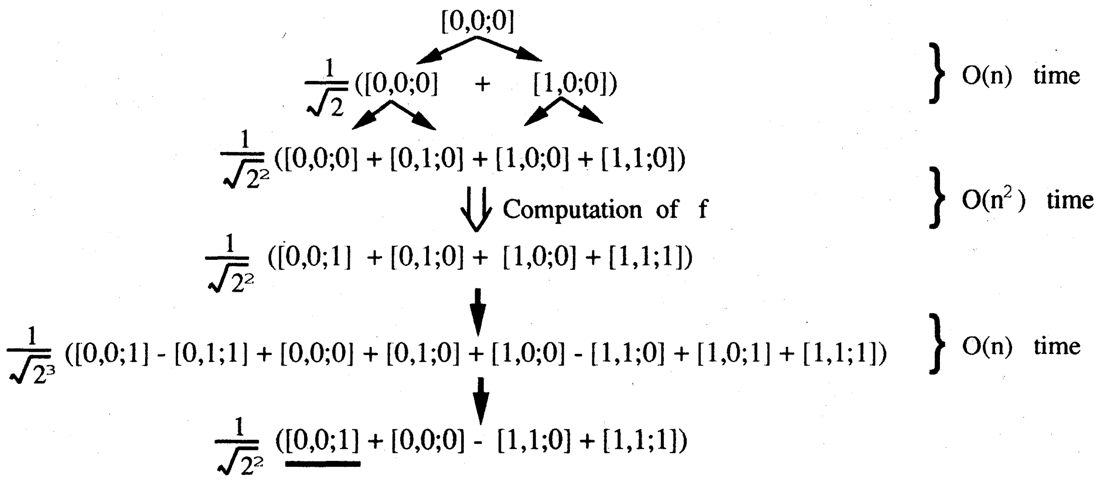
[0,0;0]
$\frac{1}{\sqrt{2}}$([0,0;0] + [1,0;0])
$\frac{1}{\sqrt{2^2}}$([0,0;0] + [0,1;0] + [1,0;0] + [1,1;0])
Computation of f
$\frac{1}{\sqrt{2^2}}$([0,0;1] + [0,1;0] + [1,0;0] + [1,1;1])
$\frac{1}{\sqrt{2^3}}$([0,0;1] - [0,1;1] + [0,0;0] + [0,1;0] + [1,0;0] - [1,1;0] + [1,0;1] + [1,1;1])
$\frac{1}{\sqrt{2^2}}$([0,0;1] + [0,0;0] - [1,1;0] + [1,1;1])
O(n) time
O(n^2) time
O(n) time

Figure 3: The change of the superposition of configurations of the UQTM U.

Acknowledgements The authors would like to express their sincere thanks to Professor Kojiro Kobayashi at the Tokyo Institute of Technology for his detailed comments on a previous version of this paper.

References

[1] Y. Aharonov, J. Anandan and L. Vaidman, “Meaning of the Wave Function”, *Phys. Rev.*, **A 47**, pp. 4616-4626, 1993.

[2] C. H. Bennett, “Logical Reversibility of Computation”, *IBM J. Res. Dev.*, **17**, pp. 525-532, 1973.

[3] P. A. Benioff, “Quantum Mechanical Hamiltonian Models of Discrete Processes That Erase Their Own Histories : Application to Turing Machines”, *Int. J. Theor. Phys.*, **21**, pp. 177-201, 1982.

[4] E. Bernstein and U. Vazirani, “Quantum Complexity Theory”, *Proc. of 25th ACM Symposium on Theory of Computing*, pp. 11-20, 1993.

[5] D. Deutsch, “Quantum Theory,the Church-Turing Principle and the Universal Quantum Computer”, *Proc. R. Soc. Lond.*, **A 400**, pp. 97-117, 1985.

[6] D. Deutsch and R. Jozsa “Rapid Solution of Problems by Quantum Computation”, *Proc. R. Soc. Lond.*, **A 439**, pp. 553-558, 1992.

[7] 三原孝志, 西野哲朗, “量子コンピュータを用いた NP 完全問題の多項式時間解法”, Research Report IS-RR-93-0012F, JAIST, 1993.