Integer Factorization and Discrete Logarithm problem are neither in P nor NP-complete

Satoshi Tazawa † University of Tokyo

1. INTRODUCTION

The function problem version of integer factorization is a problem to factorize a given positive integer $N$ into prime factors. The function problem version of discrete logarithm is a problem to find the discrete logarithm $\log_h (\text{mod } n)$. As many cryptography techniques are based on integer factorization or discrete logarithm problem, the computational complexity of these problems are crucially important to ensure the computer security[5][14].

In addition to the practical importance, these problems, together with graph isomorphism problem, are one of the candidates whose computational complexity remained unknown[6]. $P \neq NP$ problem[4][7][10], one of the most important problems in the theoretical computer science, can be settled down by showing that integer factorization and discrete logarithms are neither in P nor NP-complete.

2. ANALYSIS OF COMPUTATIONAL COMPLEXITY

2.1 Integer Factorization

In this section, to compare the other problems in NP, the decision problem version of integer factorization will be discussed. If integer factorization problem is NP-complete, other NP-complete problem can be Turing reducible to the candidate problem in polynomial time.

To check the possibility of the reduction, it is necessary to see the number of variables, parameters and invariants for each problem and to calculate the value $d$, defined as

$$d = \#\text{variables} + \#\text{parameters} - \#\text{invariants} \quad (1)$$

First, a problem to judge the existence of solution $(x, y)$ in natural number for diophantine equation

$$\alpha x^2 + \beta y - \gamma = 0 \quad (\alpha, \beta, \gamma \in \mathbb{N}) \quad (2)$$

is shown to be NP-complete[10] and it has 2 variables $(x, y)$, 3 parameters $(\alpha, \beta, \gamma)$ and 1 invariant $(2)$. Thus

$$d_{\text{NP complete}} = 2 + 3 - 1 = 4 \quad (3)$$

As all of the NP-complete problems can be reduced to $(2)$ in polynomial time, NP-complete problems such as 3-SAT with arbitrarily many variables can be reduced to $(2)$ satisfying $d_{\text{NP complete}} = 4$.

Second, the decision problem version of integer factorization is a problem to find a solution $x$ in natural number such that

$$N \equiv 0 \pmod{x} \quad (4)$$

$$1 < x < c \leq \sqrt{N} \quad (c \in \mathbb{N}) \quad (5)$$

If $c > \sqrt{N}$, this problem can be reduced to primality test for $N$ and can be solved in polynomial time[1]. So the condition $c \leq \sqrt{N}$ is reasonable to deal with integer factorization in general. $(4)$ and $(5)$ are the same as finding a solution $(x, y)$ in natural number such that

$$1 < x < c \leq \sqrt{N} \quad (c \in \mathbb{N}) \quad (6)$$
\[ c \leq y < N \] (7)
\[ xy = N \] (8)
and are the same as finding a solution \((x, y)\) in natural number such that
\[ (x - 1)(c - x) = i \ (c \leq \sqrt{N}, i \in \mathbb{N}) \] (9)
\[ (y - c)(N - y) = j \ (j \in \mathbb{N}) \] (10)
\[ xy = N \] (11)
It has 2 variables \((x, y)\), 4 parameters \((N, c, i, j)\) and 3 invariants \((9)(10)(11)\) Thus
\[ d_{\text{factorization}} = 2 + 4 - 3 = 3 \] (12)

Note that from the discussion above, the following pairs of diophantine equations
\[ (x - 1)(\sqrt{N} + 1 - x) = i \ (i \in \mathbb{N}) \] (13)
\[ (y - \sqrt{N} - 1)(N - y) = j \ (j \in \mathbb{N}) \] (14)
\[ xy = N \] (15)
are solvable for \((x, y, i, j)\) in natural number, if and only if \(N\) is composite number and are unsolvable if and only if \(N\) is a prime number. So these diophantine equations are other representation for prime numbers.

Finally, a problem \(X\) for a prime number \(p\) such that
\[ x^2 \equiv 2 \ (\text{mod } p) \] (16)
can be solved in polynomial time in both (A) and (B),
(A)If \(p = 2\), \(16\) has the trivial solutions where \(x\) is even.
(B)If \(p \neq 2\), computing the Legendre symbol
\[ \left(\frac{2}{p}\right) \equiv 2^{\frac{p-1}{2}} \ (\text{mod } p) \] (17)
by using exponentiation by squaring results in \pm 1, and \(16\) has the solution when the result is 1.
As \(16\) is the same as finding solution \(x\) such that
\[ x^2 = p \cdot q + 2 \ (q \in \mathbb{N}) \] (18)
and it has 1 variable \(x\), 2 parameters \((p, q)\) and 1 invariant, thus
\[ d_X = 1 + 2 - 1 = 2 \] (19)
To show that the reduction is impossible, the following theorem will be used.

**Theorem 1.** If \(d > d'\), it is impossible to reduce a problem \(A(d)\) to \(B (d')\) in polynomial time.

**Proof.** Take Hilbert tenth problem as a problem \(A\). Hilbert tenth problem is to test the existence of solution for
\[ P_A(x_1, x_2, \ldots, x_n) = 0 \ (P_A \text{ is polynomial}) \] (20)
\[ d_{\text{Hilbert}} > n - 1 \] (21)
and it is shown that if \(n > 9\), it’s impossible to solve\((11)(12)\). Take a problem \(B\) to find a solution \(x_1\) for
\[ P_B(x_1) = ax_1 - b = 0 \ (a, b \in \mathbb{N}) \] (22)
in polynomial time and \(22\) can be solved in polynomial time. If it is possible to reduce \(A(d)\) to \(B(d')\) in polynomial time, Hilbert tenth problem turned out to be solvable in polynomial time, but that contradicts. Therefore reducing \(A(d > n - 1)\) to \(B(d' = 1 + 2 - 1 = 2)\) is impossible in polynomial time. \(\square\)

By theorem 1, it is shown to be impossible to reduce NP-complete problem \((d = 4)\) to integer factorization problem \((d = 3)\) in polynomial time. Therefore integer factorization problem is not NP-complete. In addition, integer factorization problem \((d = 3)\) cannot be reduced to problem \((d = 2)\) in \(P\). Therefore integer factorization problem cannot be solved in polynomial time.

### 2.2 Discrete Logarithm

Discrete logarithm problem is a problem to find a solution \(x\) for \(n \in \mathbb{N}\) such that
\[ g^x \equiv h \ (\text{mod } n) \] (23)
Practically, a prime number \(p \neq 2\) is used instead of \(n\), and it is assumed that \(g^i \ (\text{mod } p) \ (i = 0, \ldots, p - 2)\) generates all residues except zero. Under such assumption, \(23\) is the same as finding solution \((x, y)\) in natural number such that
\[ g^x \equiv h \ (\text{mod } p) \] (24)
\[ h^y \equiv g \ (\text{mod } p) \] (25)
\[ xy = p - 1 \] (26)

**Theorem 2.** \(\text{DiscreteLog (mod } p) \leq_P \text{ Factorization}\)

**Proof.** As shown in \(20\), all of solutions \(x\) are the divisors of \(p - 1\). Repeat division of \(p - 1\) by 2 until it becomes odd, and the rest of the divisor of \(p - 1\) can be computed by integer factorization. After all of the divisors are obtained, discrete logarithm can also be computed in polynomial time by using exponentiation by squaring to check \(23\). \(\square\)

**Theorem 3.** \(\text{DiscreteLog (mod } n) \equiv_P \text{ Factorization and DiscreteLog (mod } p)\)

**Proof.** See \(3\). \(\square\)

By theorem 2, discrete logarithm problem (mod prime) is shown to be not NP-hard because factorization is not NP-hard. By theorem 3, discrete logarithm problem (mod composite) is neither solvable in polynomial time nor NP-hard.

Whether discrete logarithm problem (mod prime) is in \(P\) or not can not be judged from the discussion above. But under the assumption that all types of discrete logarithm problem is in the same computational complexity class, discrete logarithm problem in general is thought to be neither in \(P\) nor NP-hard. Without such assumption, integer factorization is as hard as discrete logarithm (mod composite). Therefore the decision problem version of discrete logarithm problem (mod composite) is also neither in \(P\) nor NP-complete.

### 3. \(P \neq NP\)
3.1 Proof for $P \neq NP$

By Ladner’s theorem, $P \neq NP$ if and only if the class neither in $P$ nor $NP$-complete, called $NP$-intermediate is not empty. Now that integer factorization and discrete logarithm problem (mod composite) are neither in $P$ nor $NP$-complete, $NP$-intermediate is not empty. Therefore $P \neq NP$.

3.2 Graph Isomorphism Problem

As mentioned in the introduction, little is known about graph isomorphism problem, and its variables, parameters and invariants to describe it. Even the fastest known algorithm requires exponential time and graph isomorphism problem is believed to be difficult to solve in polynomial time. So far, it is unknown whether graph isomorphism problem is in $P$ or $NP$-complete or neither. But it is already shown that if graph isomorphism problem is $NP$-complete, polynomial time hierarchy collapses to its second level. Now that $P \neq NP$ and polynomial time hierarchy does not collapse, graph isomorphism problem is not likely to be $NP$-complete (but not proved yet).

3.3 Relationship to Quantum Computers

As shown by Shor’s algorithm, integer factorization and discrete logarithm problem can be solved in polynomial time by quantum computers. Now it is shown that these problems cannot be solved in polynomial time by a classical computer, quantum computers are more powerful than a classical computer. Whether or not $NP$-complete problems can be solved by quantum computers in polynomial time remains unknown.

4. CONCLUSIONS

Integer factorization and discrete logarithm problem (mod composite) are shown to be neither in $P$ nor $NP$-complete. As a result, the conclusion, $P \neq NP$ is reached. As all of the $NP$-complete problems turned out to be impossible to solve in polynomial time by a classical computer, heuristic approaches or algorithms for restricted types of inputs need to be developed for $NP$-complete problems.

5. REFERENCES