Graph Non-Isomorphism Has a Succinct Quantum Certificate

Tatsuaki Okamoto* Keisuke Tanaka†

Summary

This paper presents the first quantum computational characterization of the Graph Non-Isomorphism problem (GNI). We show that GNI is in a quantum non-deterministic polynomial-time class, i.e., quantum Merlin-Arthur game class (QMA) or quantum probabilistic NP (BQNP). In other words, given two graphs $(G_0, G_1) \in$ GNI, (G_0, G_1) has a polynomial-size quantum certificate with which $(G_0, G_1) \in$ GNI can be convinced by a polynomial-time (probabilistic) quantum Turing machine.

1 Introduction

It is very important to characterize a natural computational problem in the light of quantum computational complexity classes. Only a few natural computational problems have been characterized in this viewpoint. For example, (decisional) problems regarding the integer factoring and discrete logarithm problems are characterized as BQP [10]. A generic problem to find a solution by an exhaustive search can be solved by a quantum computer with computational complexity of the square root of the (classical) exhaustive search complexity [6].

This paper presents the first characterization of a natural computational problem, Graph Non-Isomorphism (GNI), in the light of a quantum non-deterministic polynomial-time class, i.e., quantum Merlin-Arthur game class (QMA) or quantum probabilistic NP (BQNP) [8, 7, 11]. We show that GNI is in QMA. This result contrasts with the result that GNI (in co-NP) is not known to be in MA.

Note that there are two types of definitions for the quantum non-deterministic polynomial-time classes, QMA (or BQNP) and NQP [1, 5]. In the definition of NQP, we view non-determinism as a probabilistic process and consider whether the resulting process has zero or non-zero probability of success. NQP is defined to be the class of languages L for which there exist polynomial-time quantum Turing machines that accept with non-zero probability if and only if the input is in L.

In the definition of QMA, we view non-determinism as verification process with a (non-determistically given) certificate. Here, the certificates are quantum strings and the quantum polynomial-time verification procedure operates with two-sided bounded error. This is a quantum generalization of the class MA [2, 3], which is a probabilistic generalization of NP. (Kitaev [7] referred to the class we call QMA as BQNP.)

*NTT Laboratories, 1-1 Hikarinooka, Yokosuka-shi, Kanagawa, 239-0847 Japan. okamoto@isl.ntt.co.jp

†Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology, 1-12-1 Ookayama Meguro-ku, Tokyo 152-8552, Japan. keisuke@is.titech.ac.jp

We have developed two major key ideas in our construction of certificates for GNI. The first idea is a way to measure the distance of two vectors of quantum states, both of which are parts of superposition of a quantum string. By applying Hadamard transformation to certain bits, we can evaluate the distance between the two vectors by measuring a certain bit pattern (e.g., '10'). To check the graph non-isomorphism of graphs G_0 and G_1, the superposition, $\sum_{\pi_i \in S_n, b} |b\rangle|\pi_i(G_b)\rangle$, of all permutations of graphs G_0 and G_1 is useful. If we make such superposition by ourselves, it may always include garbage information, π_i, such as $\sum_{\pi_i \in S_n, b} |b\rangle|\pi_i(G_b)\rangle|\pi_i\rangle$. To check the validity of quantum certificate, $\sum_{\pi_i \in S_n, b} |b\rangle|\pi_i(G_b)\rangle$, we prepare the series of sub-superpositions ($j = m-1, \ldots, 1$), $\sum_{\pi_i \in S_n, b} |b\rangle|\pi_i(G_b)\rangle|[\pi_i]_j, 0 \cdots 0\rangle$, where $[x]_j$ denotes the j most significant bits of x and $|x| = m$. By using the first key idea, we can confirm that the distances between all two neighboring sub-superpositions are small, and in total we can check that the main part of the certificate, $\sum_{\pi_i \in S_n, b} |b\rangle|\pi_i(G_b)\rangle$, is close to the correct one with high probability.

2 Preliminaries

We assume the reader is familiar with the basic notions from quantum computaion. See, for example, a book by Nielsen and Chuang [9].

In this paper, we focus on the Graph Non-Isomorphism problem (GNI) defined as follows:

Instance: Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$.

Question: Are G_1 and G_2 non-isomorphic, i.e., is there no one-to-one onto function $f : V_1 \to V_2$ such that $\{u, v\} \in E_1$ if and only if $\{f(u), f(v)\} \in E_2$?

This problem is in co-NP and is not known to be polynomially solvable.

We assumu all input strings are over the alphabet $\Sigma = \{0, 1\}$. We also focus on the power of QMA defined as follows:

Definition 1 (Watrous) *A language $A \in \Sigma^*$ is in QMA if there exists a polynomial-time uniformly generated family of quantum circuits $\{Q_x\}_{x \in \Sigma^*}$ such that (i) if $x \in A$ then there exists a quantum state $|\phi\rangle$ such that $\Pr[Q_x$ accepts $|\phi\rangle] > 2/3$, and (ii) if $x \notin A$ then for all quantum states $|\phi\rangle$, $\Pr[Q_x$ accepts $|\phi\rangle] < 1/3$.*

The above definition is proposed by Watrous [11].

We denote by S_n the set of all permutations over n elements, by π_i ($i = 1, 2, \ldots, n!$) an element of S_n, and by $\pi_i(G_b)$ the graph obtained by applying permutation π_i to graph G_b.

3 Certificates

In this section, we define a set of quantum strings, which can be a quantum certificate for the Graph Non-Isomorphism problem. We denote this set as $W = \{W_1, W_2, \ldots, W_{n^12}\}$, each $W_i \in W$ is a quantum string. We require each $W_i \in W$ is the same, i.e., $W_i = W_j$ for each i, j.

We assume each $W_i \in W$ is contained in a quantum register $\mathbf{R^i} = (\mathbf{R}^i_1, \mathbf{R}^i_2, \mathbf{R}^i_3, \mathbf{R}^i_4)$. However, we require each $W_i \in W$ is the same, and will deal with each $W_i \in W$ independently. Thus, when it is clear, we will omit index i on register and write $\mathbf{R} = (\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \mathbf{R}_4)$.

Let n be the number of vertices in graphs G_0, G_1. We denote by $N = n!$ the total number of permutations on n elements. Let $m = O(n \log n)$ be the number of qubits contained in register $\mathbf{R}_4$.

Now, we define a quantum string $W_i \in W$. The first register, $\mathbf{R}_1$, represents an index of graph G_0, G_1. It consists of one qubit.

The second register, $\mathbf{R}_2$, represents an index which will be used for verification. It consists of $m+1$ qubits.

The third register, $\mathbf{R}_3$, represents the graph which is applied by a permutation corresponding to the content in register $\mathbf{R}_4$. With a reasonable representation of graph, for example, an adjacent matrix on vertices, the third register consists of $l = n(n-1)/2$ qubits.

Finally, the forth register, $\mathbf{R}_4$, contains a prefix of a representation of permutation. The other bits in register $\mathbf{R}_4$ are '$0 \cdots 0$'. With a reasonable representation of permutation, a permutation can be represented by some polynomial number of qubits as in [4].

The total number of qubits in each string $W_i \in W$ is $2m + l + 2 = O(n^2 + m)$. We denote by $[x]_y$ the y most significant bits of vector x. Then, a quantum string from set $W = \{W_1, W_2, \ldots, W_M\}$ can be depicted in Figure 1.

As mentioned in [11], a simple analysis reveals that entanglement among the quantum strings in a certificate can yield no increase in the probability of acceptance as compared to the situation in which these strings are not entangled, and that the probability of error is bounded by the tail of a binomial series as expected.

4 Verification procedure

In a quantum Merlin-Arthur game, first Merlin sends Arthur a quantum certificate, then Arthur verifies it by a quantum polynomial-time algorithm. In our protocol, Merlin sends Arthur, as a certificate, a set of quantum strings defined in the previous section. In this section, we describe the verification procedure for Arthur, which can run in polynomial-time by quantum computers. The verification procedure consists of six steps. Each step, except Step 4, decides whether we should continue the procedure or we reject an input and halt the procedure.

In Step 1, the verification procedure checks the distribution on the $\mathbf{R}_1$ register part.

Step 1:
count $\leftarrow 0$
count' $\leftarrow 0$
Randomly choose n^8 strings from W
for each chosen W_i do
    Observe the $\mathbf{R}_1$ register part (one bit)
    if this part is '0' then count $\leftarrow$ count $+1$
    if this part is '1' then count' $\leftarrow$ count $+1$
if |count−count' | $>$ $\sqrt{n^6}$ then *reject*

In Step 2, the verification procedure checks that the validity of the certificate with respect to the $\mathbf{R}_2$ register part.

$$
\begin{array}{rllll}
 & (\mathbf{R}_1) & (\mathbf{R}_2) & (\mathbf{R}_3) & (\mathbf{R}_4) \\
\frac{1}{\sqrt{2(m+1)N}}(& \sqrt{N}|0\rangle & |1000\cdots000\rangle & |000\cdots000\rangle & |000\cdots000\rangle \\
 & +|0\rangle & |0100\cdots000\rangle & |\pi_1(G_0)\rangle & |[\pi_1]_{m-1},0\rangle \\
 & +|0\rangle & |0100\cdots000\rangle & |\pi_2(G_0)\rangle & |[\pi_2]_{m-1},0\rangle \\
 & & & \vdots & \\
 & +|0\rangle & |0100\cdots000\rangle & |\pi_{N-1}(G_0)\rangle & |[\pi_{N-1}]_{m-1},0\rangle \\
 & +|0\rangle & |0100\cdots000\rangle & |\pi_N(G_0)\rangle & |[\pi_N]_{m-1},0\rangle \\
 & +|0\rangle & |0010\cdots000\rangle & |\pi_1(G_0)\rangle & |[\pi_1]_{m-2},00\rangle \\
 & +|0\rangle & |0010\cdots000\rangle & |\pi_2(G_0)\rangle & |[\pi_2]_{m-2},00\rangle \\
 & & & \vdots & \\
 & +|0\rangle & |0010\cdots000\rangle & |\pi_{N-1}(G_0)\rangle & |[\pi_{N-1}]_{m-2},00\rangle \\
 & +|0\rangle & |0010\cdots000\rangle & |\pi_N(G_0)\rangle & |[\pi_N]_{m-2},00\rangle \\
 & & & \vdots & \\
 & +|0\rangle & |0000\cdots010\rangle & |\pi_1(G_0)\rangle & |[\pi_1]_1,00\cdots00\rangle \\
 & +|0\rangle & |0000\cdots010\rangle & |\pi_2(G_0)\rangle & |[\pi_2]_1,00\cdots00\rangle \\
 & & & \vdots & \\
 & +|0\rangle & |0000\cdots010\rangle & |\pi_{N-1}(G_0)\rangle & |[\pi_{N-1}]_1,00\cdots00\rangle \\
 & +|0\rangle & |0000\cdots010\rangle & |\pi_N(G_0)\rangle & |[\pi_N]_1,00\cdots00\rangle \\
 & +|0\rangle & |0000\cdots001\rangle & |\pi_1(G_0)\rangle & |00\cdots00\rangle \\
 & +|0\rangle & |0000\cdots001\rangle & |\pi_2(G_0)\rangle & |00\cdots00\rangle \\
 & & & \vdots & \\
 & +|0\rangle & |0000\cdots001\rangle & |\pi_{N-1}(G_0)\rangle & |00\cdots00\rangle \\
 & +|0\rangle & |0000\cdots001\rangle & |\pi_N(G_0)\rangle & |00\cdots00\rangle \\
 & +\sqrt{N}|1\rangle & |1000\cdots000\rangle & |000\cdots000\rangle & |000\cdots000\rangle \\
 & +|1\rangle & |0100\cdots000\rangle & |\pi_1(G_1)\rangle & |[\pi_1]_{m-1},0\rangle \\
 & & & \vdots & \\
 & +|1\rangle & |0000\cdots001\rangle & |\pi_N(G_1)\rangle & |00\cdots00\rangle).
\end{array}
$$

Figure 1: An initial quantum string from a certificate

Step 2:
Randomly choose n^{11} strings from W, which were not chosen in Step 1
for each chosen W_i do
  Observe the $\mathbf{R}_2$ register part $(m+1$ bits$)$
  if this part does not consist of single '1' bit and m '0' bits then *reject*

In Step 3, the verification procedure checks that the validity of the certificate with respect to the $(\mathbf{R}_3, \mathbf{R}_4)$ register part when the $\mathbf{R}_2$ register part is '$100 \cdots 0$'.

Step 3:
count $\leftarrow$ 0
Randomly choose m^6 strings from W, which were not chosen thus far
in Step 1, 2
for each chosen W_i do
  Observe the $\mathbf{R}_2$ register part $(m+1$ bits$)$
  if this part is '$100 \cdots 0$' then
    count $\leftarrow$ count $+1$
    Observe the $(\mathbf{R}_3, \mathbf{R}_4)$ register part $(l+m$ bits$)$
    if this part includes '1' then *reject*
if count $<$ m^4 then *reject*

In Step 4, as mentioned in the introduction, we will makes the superposition of graphs and permutations by ourselves. In order to do this, we will use two unitary transformations, F_1 and F_2. Let F_1 be a transformation on $(\mathbf{R}_2, \mathbf{R}_4)$ register such that

$$|10\cdots 0\rangle|0\cdots 0\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{i \in S_n} |10\cdots 0\rangle|\pi_i\rangle,$$

and it does not change the contents in $\mathbf{R}_4$ register when $\mathbf{R}_2$ register is not $|10\cdots 0\rangle$. Transformation F_1 can be performed by polynomial-time quantum algorithms by [4].

Let F_2 be a transformation on $(\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \mathbf{R}_4)$ register such that

$$|b\rangle|10\ldots 0\rangle|00\cdots 0\rangle|\pi_i\rangle \rightarrow |b\rangle|10\ldots 0\rangle|\pi_i(G_b)\rangle|\pi_i\rangle,$$

and it does not change the contents in $\mathbf{R}_3$ register when $\mathbf{R}_2$ register is not $|10\cdots 0\rangle$. Transformation F_2 can be performed by polynomial-time quantum algorithms.

Step 4:
for each $W_i \in$ W, which were not chosen in Step 2 or Step 3, do
  Apply transformation F_1 to the $(\mathbf{R}_2, \mathbf{R}_4)$ register part
   to obtain the superposition of all permutations over n elements $(\sum_{i \in S_n} \ |\pi_i\ \rangle)$
   in $\mathbf{R}_4$ register when the $\mathbf{R}_2$ register part is '$100 \cdots$ 0'
  Apply transformation F_2 to the $(\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \mathbf{R}_4)$ register part
   to obtain the graph after applying the permutation in register $\mathbf{R}_3$ indicated by the index
   of G_0, G_1 in register $\mathbf{R}_1$ and the permutation in register $\mathbf{R}_4$
   when the $\mathbf{R}_2$ register part is '$100 \cdots$ 0'

$$
\begin{array}{rllll}
 & (\mathbf{R}_1) & (\mathbf{R}_2) & (\mathbf{R}_3) & (\mathbf{R}_4) \\
\frac{1}{\sqrt{2(m+1)N}}(& |0\rangle & |1000\cdots000\rangle & |\pi_1(G_0)\rangle & |\pi_1\rangle \\
+ & |0\rangle & |1000\cdots000\rangle & |\pi_2(G_0)\rangle & |\pi_2\rangle \\
 & & & \vdots & \\
+ & |0\rangle & |1000\cdots000\rangle & |\pi_{N-1}(G_0)\rangle & |\pi_{N-1}\rangle \\
+ & |0\rangle & |1000\cdots000\rangle & |\pi_N(G_0)\rangle & |\pi_N\rangle \\
+ & |0\rangle & |0100\cdots000\rangle & |\pi_1(G_0)\rangle & |[\pi_1]_{m-1},0\rangle \\
+ & |0\rangle & |0100\cdots000\rangle & |\pi_2(G_0)\rangle & |[\pi_2]_{m-1},0\rangle \\
 & & & \vdots & \\
+ & |0\rangle & |0100\cdots000\rangle & |\pi_{N-1}(G_0)\rangle & |[\pi_{N-1}]_{m-1},0\rangle \\
+ & |0\rangle & |0100\cdots000\rangle & |\pi_N(G_0)\rangle & |[\pi_N]_{m-1},0\rangle \\
+ & |0\rangle & |0010\cdots000\rangle & |\pi_1(G_0)\rangle & |[\pi_1]_{m-2},00\rangle \\
+ & |0\rangle & |0010\cdots000\rangle & |\pi_2(G_0)\rangle & |[\pi_2]_{m-2},00\rangle \\
 & & & \vdots & \\
+ & |0\rangle & |0010\cdots000\rangle & |\pi_{N-1}(G_0)\rangle & |[\pi_{N-1}]_{m-2},00\rangle \\
+ & |0\rangle & |0010\cdots000\rangle & |\pi_N(G_0)\rangle & |[\pi_N]_{m-2},00\rangle \\
 & & & \vdots & \\
+ & |0\rangle & |0000\cdots010\rangle & |\pi_1(G_0)\rangle & |[\pi_1]_1,00\cdots00\rangle \\
+ & |0\rangle & |0000\cdots010\rangle & |\pi_2(G_0)\rangle & |[\pi_2]_1,00\cdots00\rangle \\
 & & & \vdots & \\
+ & |0\rangle & |0000\cdots010\rangle & |\pi_{N-1}(G_0)\rangle & |[\pi_{N-1}]_1,00\cdots00\rangle \\
+ & |0\rangle & |0000\cdots010\rangle & |\pi_N(G_0)\rangle & |[\pi_N]_1,00\cdots00\rangle \\
+ & |0\rangle & |0000\cdots001\rangle & |\pi_1(G_0)\rangle & |00\cdots00\rangle \\
+ & |0\rangle & |0000\cdots001\rangle & |\pi_2(G_0)\rangle & |00\cdots00\rangle \\
 & & & \vdots & \\
+ & |0\rangle & |0000\cdots001\rangle & |\pi_{N-1}(G_0)\rangle & |00\cdots00\rangle \\
+ & |0\rangle & |0000\cdots001\rangle & |\pi_N(G_0)\rangle & |00\cdots00\rangle \\
+ & |1\rangle & |1000\cdots000\rangle & |\pi_1(G_1)\rangle & |\pi_1\rangle \\
+ & |1\rangle & |1000\cdots000\rangle & |\pi_2(G_1)\rangle & |\pi_2\rangle \\
 & & & \vdots & \\
+ & |1\rangle & |1000\cdots000\rangle & |\pi_{N-1}(G_1)\rangle & |\pi_{N-1}\rangle \\
+ & |1\rangle & |1000\cdots000\rangle & |\pi_N(G_1)\rangle & |\pi_N\rangle \\
+ & |1\rangle & |0100\cdots000\rangle & |\pi_1(G_1)\rangle & |[\pi_1]_{m-1},0\rangle \\
 & & & \vdots & \\
+ & |1\rangle & |0000\cdots001\rangle & |\pi_N(G_1)\rangle & |00\cdots00\rangle).
\end{array}
$$

Figure 2: A quantum string from a certificate after Step 4

After Step 4, string $W_i \in W$ which were not chosen in Step 1, 2, 3 is shown in Figure 2.

In Step 5, as mentioned in the introduction, we will check that the distance between each two neighboring sub-superpositions is small.

Step 5:
for each $j = 1, 2, \ldots, m$ do
  count $\leftarrow$ 0
  Randomly choose m^8 strings from W, which were not chosen thus far in Step 1, 2, 3, 5
  for each chosen W_i do
    Observe the $\mathbf{R}_1$ register part (one bit) (say b)
    Observe the $(j-1)$ most significant bits and the $(m-j)$ least significant bits of the $\mathbf{R}_2$ register part
    if these bits are all '0' then
      count $\leftarrow$ count $+1$
      Apply transformation F_3 to j-th and $(j+1)$-th bit of $\mathbf{R}_2$ register part such that $|00\rangle \rightarrow |10\rangle$, $|01\rangle \rightarrow |01\rangle$, $|10\rangle \rightarrow |00\rangle$, $|11\rangle \rightarrow |11\rangle$
      Apply Hadamard transformation to $(j+1)$-th bit of the $\mathbf{R}_2$ register part and to the $(m-j+1)$-th bit of the $\mathbf{R}_4$ register part
      Observe $(j+1)$-th bit of the $\mathbf{R}_2$ register part and $(m-j+1)$-th bit of the $\mathbf{R}_4$ register part
      if these bits are '10' then *reject*
  if count $<$ m^5 then *reject*

Finally, in Step 6, we perform Hadamard transformation on the first register to estimate the orthogonality on the third register including graphs.

Step 6:
count $\leftarrow$ 0
count$'$ $\leftarrow$ 0
Randomly choose m^3 strings from W, which were not chosen thus far in Step 1, 2, 3, 5
For each chosen W_i
  Observe the $(\mathbf{R}_2, \mathbf{R}_4)$ register part
  if the $\mathbf{R}_2$ register part is '$0 \cdots 01$' and the $\mathbf{R}_4$ register part is '$0 \cdots 0$' then
    count $\leftarrow$ count $+1$
    Apply Hadamard transformation to the $\mathbf{R}_1$ register part and observe this part
    if '1' is observed then count$'$ $\leftarrow$ count$'$ $+1$
if count$'$ /count $<$ $m/3$ then *reject*
if the verification procedure has not rejected thus far, then *accept*

5 Proof of completeness

In this section and the next section, we prove that the Graph Non-Isomorphism problem is in QMA. First, in this section, we show that two non-isomorphic input graphs can be accepted with a certificate with high probability.

Assume that graph G_0 and G_1 are non-isomorphic. In this case we must prove that there exists a certificate causing the verification procedure to accept with high probability. The certificate will be W described in Section 2.

In Step 1, the verification procedure checks the validity of the certificate. By applying the Chernoff bound, the probability that the verification procedure rejects in this step is bounded by $O(e^{-n^4})$.

In Step 2 and Step 3, the verification procedure checks the validity of the certificate. The probability that the verification procedure rejects in these steps is therefore 0.

In Step 4, the verification procedure only performs transformation F_1 and F_2 described above. Thus, this step cannot decrease the probability of acceptance.

In Step 5, the verification procedure checks the validity of the certificate. The probability that the verification procedure rejects in this step is therefore 0.

Finally, in Step 6, since graph G_0 and G_1 are non-isomorphic, the probability that '1' is observed in $\mathbf{R}_1$ register after applying Hadamard transformation is 1/2. Thus, the probability that the verification procedure rejects in this step is exponentially small. Therefore, the verification procedure accepts with overwhelming probability ($> 2/3$).

6 Proof of soundness

Now suppose graphs G_0 and G_1 are isomorphic. In this case we must bound the probability of acceptance by 1/3.

In Step 1, the verification procedure checks the validity of the distribution of the $\mathbf{R}_1$ register values of the certificate strings in W. By applying the Chernoff bound, the ratio of '1' in the superposition of the $\mathbf{R}_1$ register values of the strings in W is lower (upper) bounded by $1/2 - 4/n^2(1/2 + 4/n^2)$ with overwhelming probability ($> 1 - e^{-n^4}$).

In Step 2, the verification procedure checks the validity of the $\mathbf{R}_2$ register values of the certificate strings in W. Suppose that a $1/n^{10}$ fraction of the strings in W are invalid with respect to the $\mathbf{R}_2$ register value. Then, the probability that Step 2 cannot detect invalid certificates is at most $(1 - 1/n^{10})^{n^{11}} \leq e^{-n}$. Therefore, at least $(1 - 1/n^{10})$ fraction of the of the strings in W are valid regarding the $\mathbf{R}_2$ register value with overwhelming probability ($> 1 - e^{-n}$).

In Step 3, the verification procedure checks the validity of the $(\mathbf{R}_3, \mathbf{R}_4)$ register values of the certificate strings with $\mathbf{R}_2 =$ '$100 \cdots 0$' in W. Suppose that $1/m^3$ of the strings in W such that '$100 \cdots 0$' is observed in $\mathbf{R}_2$ register are invalid with respect to the $(\mathbf{R}_3, \mathbf{R}_4)$ register part. Then, the probability that Step 3 cannot detect invalid certificates is at most $(1 - 1/m^3)^{m^4} \leq e^{-m} \leq e^{-n}$. That is, at least $(1 - 1/m^3)$ fraction of the strings with $R_2 =$ '$100 \cdots 0$' in W are valid regarding $(\mathbf{R}_3, \mathbf{R}_4)$ register values with overwhelming probability ($> 1 - e^{-n}$).

We now define the distance between two vectors, $a = (\alpha_1, \ldots, \alpha_M)$ and $b = (\beta_1, \ldots, \beta_M)$ by

$$d(a, b) = (|\alpha_1 - \beta_1|^2 + \cdots + |\alpha_M - \beta_M|^2)^{1/2},$$

where α_i and β_i are complex numbers.

Let $(z_1, \ldots, z_M)$ be orthogonal basis for quantm strings. Let

$$x = \sum_{i=1}^{M} \alpha_i |0\rangle |d_i\rangle |z_i\rangle + \sum_{i=1}^{M} \beta_i |1\rangle |0\rangle |z_i\rangle,$$

where $d_i \in \{0,1\}$. If we apply Hadamard transformation to the first two bits of x, and observe these bits, then

$$4 \times \Pr[\text{`10' is observed}] = d(a,b)^2.$$

By using this relation, we can bound the distance between two vectors by the probability of observing '01' after applying Hadamard transformation.

Step 5 verifies the distances between $(\mathbf{R}_3, \mathbf{R}_4)$ register sub-superpositions in a string. Step 2 verification guarantees that at least $(1 - 1/n^{10})$ fraction of the of the strings in W are valid regarding the $\mathbf{R}_2$ register value with overwhelming probability. So, when Step 5 selects m^8 $(= O(n^8(\log n)^8))$ strings, all of them are valid regarding the $\mathbf{R}_2$ register value with probability at least $(1 - 1/n)$. Then, by Step 5 verification, for all $j \in \{1, 2, \ldots, m\}$, $\Pr[\text{`10' is observed}]$ is bounded by $1/m^4$ with overwhelming probability $(> 1 - e^{-n})$ (by a similar evaluation as above). Therefore, the distance regarding two vectors of sub-superpositions is bounded by $2/m^2$.

We then imaginarily change the basis of $\mathbf{R}_3$ and $\mathbf{R}_4$ as follows: For $|x\rangle|y\rangle$, if there exists (π_i, j) such that $x = \pi_i(G_b)$ and $y = [\pi_i]_j$, then change basis $|x\rangle|y\rangle$ to $|x\rangle|0\cdots0\rangle$. If $\mathbf{R}_2 \neq \text{`}0\cdots01\text{'}$, $x = \pi_i(G_b)$ and $y = \text{`}0\cdots0'$, then change $|x\rangle|y\rangle$ to $|x\rangle|\pi_i\rangle$. Otherwise, do not change it. Under this imaginary basis change, the vector representation does not change, i.e., the distance between two vectors holds.

Therefore, by using the property of distance, the distance between the (base changed) $(\mathbf{R}_3, \mathbf{R}_4)$-vector with $R_2 = \text{`}100\cdots0\text{'}$ and the $(\mathbf{R}_3, \mathbf{R}_4)$-vector with $\mathbf{R}_2 = \text{`}00\cdots01\text{'}$ is bounded by $2/m$ with overwhelming probability.

Using the evaluation for Step 3, the distance between the $(\mathbf{R}_3, \mathbf{R}_4)$-vector with $\frac{1}{\sqrt{N}} \sum_{i \in S_n} |\pi_i(G_b)\rangle|0\cdots0\rangle$ and the (base changed) $(\mathbf{R}_3, \mathbf{R}_4)$-vector with $\mathbf{R}_2 = \text{`}100\cdots0$ is bounded by $1/m$ with overwhelming probability.

Therefore, finally, the distance between the $(\mathbf{R}_3, \mathbf{R}_4)$-vector with $\frac{1}{\sqrt{N}} \sum_{i \in S_n} |\pi_i(G_b)\rangle|0\cdots0\rangle$ and the $(\mathbf{R}_3, \mathbf{R}_4)$-vector with $R_2 = \text{`}00\cdots01\text{'}$ is bounded by $3/m$ with overwhelming probability.

Finally, combining the evaluation for Step 1 and the distance between the above-mentioned vectors, the probability of observing '1' in Step 6 is bounded by $10/m^2$ with overwhelming probability. Therefore, in Step 6, it is rejected with probability at most $1/n$.

7 Conclusion

This paper presented the first quantum computational characterization of the graph non-isomorphism (GNI). We showed that GNI is in a quantum non-deterministic polynomial-time class, QMA. This result is the first step to characterize natural computational problems by quantum non-determinism.

References

[1] Adleman, L. M., Demarrais, J., and Huang, M.-D. A. Quantum computability. *SIAM Journal on Computing 26*, 5 (Oct. 1997), 1524–1540.

[2] Babai, L. Trading group theory for randomness. In *Proceedings of the seventeenth annual ACM Symposium on Theory of Computing, Providence, Rhode Island, May 6–8, 1985* (1985), pp. 421–429.

[3] Babai, L., and Moran, S. Arthur-Merlin games: A randomized proof system, and a hierarchy of complexity classes. *Journal of Computer and System Sciences 36* (1988), 254–276.

[4] Beals, R. Quantum computation of Fourier transforms over symmetric groups. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing* (El Paso, Texas, 4–6 May 1997), pp. 48–53.

[5] Fenner, S., Green, F., Homer, S., and Pruim, R. Determining acceptance possibility for a quantum computation is hard for the polynomial hierarchy. *Proceedings of the Royal Society, London A 455* (1999), 3953–3966.

[6] Grover, L. K. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, May 22–24, 1996* (1996), pp. 212–219.

[7] Kitaev, A. Quantum NP. Talk at AQIP'99: Second Workshop on Algorithms in Quantum Information Processing, DePaul University, Jan. 1999.

[8] Knill, E. Quantum randomness and nondeterminism. quant-ph/9610012, 1996.

[9] Nielsen, M. A., and Chuang, I. L. *Quantum Computation and Quantum Information*. Cambridge, 2000.

[10] Shor, P. W. Algorithms for quantum computation: Discrete log and factoring. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (Nov. 1994), pp. 124–134.

[11] Watrous, J. Succinct quantum proofs for properties of finite groups. In *41st Annual Symposium on Foundations of Computer Science: proceedings: 12–14 November, 2000, Redondo Beach, California* (2000), pp. 537–546.