

Quantum Algorithm for Longest Path Problem by Hybrid Method of Grover's Database Search and Shor's Data Decrease

Toru Fujimura

Third area security office, University of Tsukuba, Ibaraki-branch, Rising Sun Security Service Co., Ltd., 1-1-1, Tennodai, Tsukuba, Ibaraki 305-8577, Japan

Abstract

A quantum algorithm for the longest path problem by a hybrid method of Grover's database search and Shor's data decrease and its example are reported. When a graph has n vertexes and m edges, the longest path is decided on turning round n vertexes with fixing a starting vertex. The computational complexity of the classical computation is $(n - 1)!/2$, because the counter routes are excluded. The computational complexity becomes about $3(\log_2(n - 1))^2(n - 1)^2$ by this quantum algorithm. Therefore, a decreased process becomes possible.

Keywords: Quantum algorithm, longest path problem, Grover's database search, Shor's data decrease, computational complexity.

AMS subject classification: Primary 81-08; Secondary 68R10, 68W40.

1. INTRODUCTION

Susa, Yamashiro, Yamamoto, and Nishimori showed an exponential speedup of the quantum annealing for a quantum computer [1]. The algorithms of the quantum computer by Deutsch-Jozsa, Shor, Grover, and so on are known [2-7]. Ambainis's

quantum walk algorithms was the example to decrease the computational complexity [8]. When the feature of the problem isn't used, it is difficult to decrease the computational complexity. Bennett, Bernstein, Brassard, and Vazirani addressed the class NP cannot be solved on a quantum Turing machine in time $O(2^{n/2})$ [9]. However, they didn't eliminate the unnecessary data on the machine's way to the end. For this reason, Fujimura suggested that the probability amplitudes of the 3-SAT problem are converged quickly by a hybrid method of Grover's database search and Shor's data decrease [3, 5-7, 10]. Its computational complexity is decreased. The longest path problem [11] is examined by the hybrid method this time. Therefore, its result is reported.

2. LONGEST PATH PROBLEM

When a graph has n vertexes and m edges, the longest path is decided on turning round n vertexes with fixing a starting vertex [11]. The computational complexity of the classical computation is $(n - 1)!/2$, because the counter routes are excluded.

3. QUANTUM ALGORITHM

It is assumed that n vertexes of $P_0(x_0, y_0)$, $P_1(x_1, y_1)$, $P_2(x_2, y_2)$, \dots , $P_{n-2}(x_{n-2}, y_{n-2})$, and $P_{n-1}(x_{n-1}, y_{n-1})$ are set [x_i and y_i are the two dimensional coordinates. $0 \leq i \leq n-1$. i is an integer.], P_0 is fixed, and a distance between P_i and P_j is $L(i, j) [= L(j, i)]$. Therefore, path of P_1, P_2, \dots, P_{n-2} , and P_{n-1} is considered.

(1) The number of the repeated permutation of $n - 1$ vertexes is $(n - 1)^{n-1}$.

(2) The number of permutation of $n - 1$ vertexes is $(n - 1)!$.

When $n - 1$ vertexes are $P_{1+a(1)}$, $P_{1+a(2)}$, \dots , $P_{1+a(n-2)}$, and $P_{1+a(n-1)}$ [where, $a(q) = a_q$; $0 \leq a_q \leq n - 2$. a_q is an integer.], it is assumed that $U[X] = a_1(n - 1)^{n-2} + a_2(n - 1)^{n-3} + \dots + a_{n-2}(n - 1)^1 + a_{n-1}(n - 1)^0$ [X is the number of datum.] is the numbering datum from 0 to $(n - 1)^{n-1} - 1$ [For example, $U[X = 0]$ is $a_1 = 0, a_2 = 0, \dots, a_{n-2} = 0$, and $a_{n-1} = 0$, and $U[X = (n - 1)^{n-1} - 1]$ is $a_1 = n - 2, a_2 = n - 2, \dots, a_{n-2} = n - 2$, and $a_{n-1} = n - 2$.] in (1).

In (2), it is assumed that the first datum $V(Y = 1)$ is $a_1 = 0, a_2 = 1, \dots, a_{n-2} = n - 3$, and $a_{n-1} = n - 2$, and the $(n - 1)!$ -th datum $V(Y = (n - 1)!) is $a_1 = n - 2, a_2 = n - 3, \dots, a_{n-2} = 1$, and $a_{n-1} = 0$, Y [where, $1 \leq Y \leq (n - 1)! - 1$. Y is an integer.] is obtained from $v_1(n - 2)! +$$

$v_2(n-3)! + \dots + v_{n-2}1!$. [where, $0 \leq v_i \leq n - (1 + i)$. v_i is an integer.] Each of t_i [$1 \leq i \leq n - 1$. i is an integer.] is 1 piece of permutation from 0 to $n - 2$.

- (I) When v_i is 0 from $i = 1$ to $i = n - 3$ sequentially, t_i is the smallest number in remained numbers.
- (II) When v_i isn't 0 from $i = 1$ to $i = n - 3$ sequentially, and $v_{i+1}, v_{i+2}, \dots, v_{n-3}$, and v_{n-2} are 0, t_i is the v_i -th small number in remained numbers, and $t_{i+1} > t_{i+2} > \dots > t_{n-2} > t_{n-1}$ is selected in remained numbers.
- (III) When v_i isn't 0 from $i = 1$ to $i = n - 3$ sequentially, and there are $v_{i+1} \neq 0$ or $v_{i+2} \neq 0$ or \dots or $v_{n-3} \neq 0$ or $v_{n-2} \neq 0$, t_i is the $(v_i + 1)$ -th small number in remained numbers.
- (IV) When v_{n-2} is 1, $t_{n-2} < t_{n-1}$ is selected in remained numbers. Therefore, $t_1(n-1)^{n-2} + t_2(n-1)^{n-3} + \dots + t_{n-2}(n-1)^1 + t_{n-1}(n-1)^0$ is $U(V(Y))$. [where, t_i is equal to a_i in (2).]

g is the minimum integer that follows $(n-1)!/2 \leq 2^{2g} = 4^g$, because a number of combinations of answers is at least 2. $U(V(Y=1))$, $U(V(Y=((n-1)!/4) - 2))$, $U(V(Y=((n-1)!/16) - 2))$, \dots , $U(V(Y=((n-1)!/4^{g-1}) - 2))$, and $U(V(Y=(n-1)!/4^g))$ are computed. [See Appendix-1] M_1 that is a starting distance value is decided at random. Next, a quantum algorithm is shown as the following.

First of all, quantum registers $|a_1\rangle, |a_2\rangle, \dots, |a_{n-1}\rangle, |b\rangle, |c_0\rangle, |c_1\rangle, \dots, |c_{n-3}\rangle, |d_1\rangle, |d_2\rangle, |e\rangle$, and $|k\rangle$ are prepared. When F is the minimum integer that is $\log_2(n-1)$ or more, each of $|a_h\rangle$ that h is an integer from 1 to $n-1$ is consisted of F quantum bits [= qubits]. States of $|a_1\rangle, |a_2\rangle, \dots, |a_{n-1}\rangle, |b\rangle, |c_0\rangle, |c_1\rangle, \dots, |c_{n-3}\rangle, |d_1\rangle, |d_2\rangle, |e\rangle$, and $|k\rangle$ are $a_1, a_2, \dots, a_{n-1}, b, c_0, c_1, \dots, c_{n-3}, d_1, d_2, e$, and k , respectively.

Step 1: Each qubit of $|a_1\rangle, |a_2\rangle, \dots, |a_{n-1}\rangle, |b\rangle, |c_0\rangle, |c_1\rangle, \dots, |c_{n-3}\rangle, |d_1\rangle, |d_2\rangle, |e\rangle$, and $|k\rangle$ is set $|0\rangle$.

Step 2: The Hadamard gate \boxed{H} acts on each qubit of $|a_1\rangle, |a_2\rangle, \dots, |a_{n-2}\rangle$, and $|a_{n-1}\rangle$ [3, 4]. It changes them for entangled states. The total states are $(2^F)^{n-1}$. [$|a_h\rangle$ is consisted of F qubits. Each qubit is acted on by \boxed{H} . Therefore, $F(n-1)$ of \boxed{H} are necessary.]

Step 3: It is assumed that a quantum gate (A) changes $|b\rangle$ for $|1\rangle$ in $a_h < n-1$, or it changes $|b\rangle$ for $|0\rangle$ in the others of a_h . As a target state for $|b\rangle$ is 1, quantum phase inversion gates (PI) and quantum inversion about mean gates (IM) act on $|b\rangle$ [Grover's database search. The same gates action is shown in the following.] [3, 6, 7].

When Q is the minimum even integer that is $(2^F/(n-1))^{1/2}$ or more, the total number that (PI) and (IM) act on $|b\rangle$ is Q , because they are a couple. Next, an observation gate (OB) observes $|b\rangle$ [Shor's data decrease. The same gate action is shown in the following.] [3, 5]. These actions are repeated sequentially from $|a_1\rangle$ to $|a_{n-1}\rangle$. Therefore, each state of $|a_h\rangle$ is 0, 1, \dots , $n-3$, and $n-2$, and the total states become $(n-1)^{n-1}$.

Step 4: It is assumed that a quantum gate (B) changes $|c_0\rangle, |c_1\rangle, \dots, |c_{n-4}\rangle$, and $|c_{n-3}\rangle$ for $|c_0+1\rangle, |c_1+1\rangle, \dots, |c_{n-4}+1\rangle$, and $|c_{n-3}+1\rangle$ in $a_h = 0, 1, \dots, n-4$, and $n-3$, respectively. This action repeats from $|a_1\rangle$ to $|a_{n-1}\rangle$. As the target state for $|c_0\rangle$ is 1, (PI) and (IM) act on $|c_0\rangle$. When R_0 is the minimum even integer that is $((n-1)/(n-2))^{(n-2)/2}$ or more, the total number that (PI) and (IM) act on $|c_0\rangle$ is R_0 . Next, (OB) observes $|c_0\rangle$. Therefore, only the paths that contain 1 piece of 0 remain. The number of data is $(n-1)(n-2)^{n-2}$. As the target state for $|c_1\rangle$ is 1, (PI) and (IM) act on $|c_1\rangle$. When R_1 is the minimum even integer that is $((n-2)/(n-3))^{(n-3)/2}$ or more, the total number that (PI) and (IM) act on $|c_1\rangle$ is R_1 . Next, (OB) observes $|c_1\rangle$. Therefore, only the paths that contain 1 piece of 1 remain. The number of data is $(n-1)(n-2)(n-3)^{n-3}$. Similarly, these actions are repeated sequentially from $|c_2\rangle$ to $|c_{n-3}\rangle$. Only the paths that contain 1 piece of number from 0 to $n-2$, respectively, remain. The number of data is $(n-1)!$ [= W_0].

Step 5: It is assumed that a quantum gate (C_1) changes $|d_1\rangle$ and $|d_2\rangle$ for $|L(0, 1+a_1) + L(1+a_1, 1+a_2)\rangle$ and $|(n-1)^{n-2}a_1 + (n-1)^{n-3}a_2\rangle$, respectively, from $|a_1\rangle$ and $|a_2\rangle$. Similarly, (C_i) [$2 \leq i \leq n-3$. i is an integer.] changes $|d_1\rangle$ and $|d_2\rangle$ for $|d_1 + L(1+a_i, 1+a_{i+1})\rangle$ and $|d_2 + (n-1)^{n-(i+2)}a_{i+1}\rangle$, respectively, from $|a_i\rangle$ and $|a_{i+1}\rangle$. This action is repeated sequentially from $|a_2\rangle$ to $|a_{n-2}\rangle$. (C_{n-2}) changes $|d_1\rangle$ and $|d_2\rangle$ for $|d_1 + L(1+a_{n-2}, 1+a_{n-1}) + L(1+a_{n-1}, 0)\rangle$ and $|d_2 + (n-1)^0a_{n-1}\rangle$, respectively, from $|a_{n-2}\rangle$ and $|a_{n-1}\rangle$. Therefore, $|d_1\rangle$ and $|d_2\rangle$ become $|L_{\text{total}} = L(0, 1+a_1) + L(1+a_1, 1+a_2) + \dots + L(1+a_{n-2}, 1+a_{n-1}) + L(1+a_{n-1}, 0)\rangle$ and $|U[X]\rangle$, respectively.

Step 6: It is assumed that a quantum gate (D) changes $|e\rangle$ for $|e+d_1\rangle$ in $d_1 \geq M_1$, or it changes $|e\rangle$ for $|e+M_1+d_2\rangle$ in the others of d_1 .

Step 7: It is assumed that a quantum gate (E_1) changes $|k\rangle$ for $|1\rangle$ in $M_1 \leq e \leq M_1 + U(V(Y = ((n-1)!/4) - 2))$, or it changes $|k\rangle$ for $|0\rangle$ in the others of e . As the target state for $|k\rangle$ is 1, (PI) and (IM) act on $|k\rangle$. The number of the data that is included in $M_1 \leq e \leq M_1 + U(V(Y = ((n-1)!/4) - 2))$ is $W_1 \approx (n-1)!/4$. [See Appendix-2] When T_1 is the minimum even integer that is $(W_0/W_1)^{1/2}$ or more, the total number that (PI) and (IM)

act on $|k\rangle$ is $T_1 \approx 2$. Next, (OB) observes $|k\rangle$, and the data of W_1 remain. Similarly, (E_i) [$2 \leq i \leq g - 1$. i is the integer.] changes $|k\rangle$ for $|1\rangle$ in $M_1 \leq e \leq M_1 + U(V(Y = ((n - 1)!/4^i) - 2))$, or it changes $|k\rangle$ for $|0\rangle$ in the others of e . As the target state for $|k\rangle$ is 1, (PI) and (IM) act on $|k\rangle$. The number of the data that is included in $M_1 \leq e \leq M_1 + U(V(Y = ((n-1)!/4^i) - 2))$ is $W_i \approx (n - 1)!/4^i$. When T_i is the minimum even integer that is $(W_{i-1}/W_i)^{1/2}$ or more, the total number that (PI) and (IM) act on $|k\rangle$ is $T_i \approx 2$. Next, (OB) observes $|k\rangle$, and the data of W_i remain. These actions are repeated sequentially from 2 to $g - 1$ at i . (E_g) changes $|k\rangle$ for $|1\rangle$ in $M_1 \leq e \leq M_1 + U(V(Y = (n - 1)!/4^g))$, or it changes $|k\rangle$ for $|0\rangle$ in the others of e . As the target state for $|k\rangle$ is 1, (PI) and (IM) act on $|k\rangle$. The number of the data that is included in $M_1 \leq e \leq M_1 + U(V(Y = (n - 1)!/4^g))$ is $W_g \approx 2$. When T_g is the minimum even integer that is $(W_{g-1}/W_g)^{1/2}$ or more, the total number that (PI) and (IM) act on $|k\rangle$ is $T_g \approx 2$. Next, (OB) observes $|a_1\rangle, |a_2\rangle, \dots, |a_{n-1}\rangle, |b\rangle, |c_0\rangle, |c_1\rangle, \dots, |c_{n-3}\rangle, |d_1\rangle, |d_2\rangle, |e\rangle$, and $|k\rangle$, and one of the data of W_g remains. Therefore, one example of paths that are $L_{\text{total}} \geq M_1$ is obtained.

Step 8: When the state of $|k\rangle$ is 1 or 0, M_1 is assumed to be $M_2 [> M_1]$ or $M_2 [< M_1]$, respectively, these computations from step 1 to step 8 are repeated. It is assumed that the maximum distance M_{max} obtains by repeating about $\log_2 (n - 1)!$ [12].

An example is shown as the next section. However, this algorithm is applied as far as the effect of Grover's database search and Shor's data decrease.

4. NUMERICAL COMPUTATION

It is assumed that there are $n = 10, m = 26, P_0(0, 0), P_1(1, -2), P_2(3, -1), P_3(4, 1), P_4(2, 3), P_5(1, -1), P_6(3, -2), P_7(4, 0), P_8(0, 1), P_9(2, 2), L(0, 2) \approx 3.2, L(0, 5) \approx 1.4, L(0, 8) = 1, L(0, 3) \approx 4.1, L(0, 7) = 4, L(1, 5) = 1, L(1, 4) \approx 5.1, L(2, 6) = 1, L(3, 7) = 1, L(3, 1) \approx 4.2, L(3, 9) \approx 2.2, L(4, 9) = 1, L(4, 6) \approx 5.1, L(4, 2) \approx 4.1, L(5, 9) \approx 3.2, L(6, 1) = 2, L(6, 8) \approx 4.2, L(7, 2) \approx 1.4, L(7, 5) \approx 3.2, L(7, 8) \approx 4.1, L(8, 4) \approx 2.8, L(8, 7) \approx 4.1, L(8, 3) = 4, L(9, 3) \approx 2.2, L(9, 2) \approx 3.2, L(9, 4) = 1$ [The value of the others of $L(i, j)$ is 0.], $g = 9$ [$9!/2 = 181440 \leq 4^9 = 262144$], $U(V(Y = 1)) = 6053444, U(V(Y = (9!/4) - 2 = 90718)) = 95584572$ [for example, $Y = 90718 = 2 \cdot 8! + 1 \cdot 7! + 6 \cdot 6! + 5 \cdot 5! + 4 \cdot 4! + 3 \cdot 3! + 2 \cdot 2! + 0 \cdot 1!$, $U(V(Y = 90718)) = 2 \cdot 9^8 + 1 \cdot 9^7 + 8 \cdot 9^6 + 7 \cdot 9^5 + 6 \cdot 9^4 + 5 \cdot 9^3 + 3 \cdot 9^2 + 4 \cdot 9^1 + 0 \cdot 9^0$], $U(V(Y = (9!/16) - 2 = 22678)) = 26275564, U(V(Y = (9!/64) - 2 = 5668)) = 10598756, U(V(Y = (9!/256) - 2 \approx 1416)) = 6894596, U(V(Y = (9!/1024) - 2 \approx 352)) = 6198348, U(V(Y = (9!/4096) - 2 \approx 87)) = 6073748, U(V(Y =$

$(9!/16384) - 2 \approx 20)) = 6055548$, $U(V(Y = (9!/65536) - 2 \approx 4)) = 6053532$, and $M_1 = 20$.

First of all, $|a_1\rangle, |a_2\rangle, \dots, |a_9\rangle, |b\rangle, |c_0\rangle, |c_1\rangle, \dots, |c_7\rangle, |d_1\rangle, |d_2\rangle, |e\rangle$, and $|k\rangle$ are prepared. When F is the minimum integer that is $\log_2(n - 1) = \log_2 9 \approx 3.170 \leq 4 = F$, each of $|a_h\rangle$ that h is the integer from 0 to 8 is consisted of $F = 4$ qubits. States of $|a_1\rangle, |a_2\rangle, \dots, |a_9\rangle, |b\rangle, |c_0\rangle, |c_1\rangle, \dots, |c_7\rangle, |d_1\rangle, |d_2\rangle, |e\rangle$, and $|k\rangle$ are $a_1, a_2, \dots, a_9, b, c_0, c_1, \dots, c_7, d_1, d_2, e$, and k , respectively.

Step 1: Each qubit of $|a_1\rangle, |a_2\rangle, \dots, |a_9\rangle, |b\rangle, |c_0\rangle, |c_1\rangle, \dots, |c_7\rangle, |d_1\rangle, |d_2\rangle, |e\rangle$, and $|k\rangle$ is set $|0\rangle$.

Step 2: $\boxed{\mathbb{H}}$ acts on each qubit of $|a_1\rangle, |a_2\rangle, \dots, |a_8\rangle$, and $|a_9\rangle$. It changes them for entangled states. The total states are $(2^F)^{n-1} = (2^4)^9$.

Step 3: (A) changes $|b\rangle$ for $|1\rangle$ in $a_h < 9$, or it changes $|b\rangle$ for $|0\rangle$ in the others of a_h . As the target state for $|b\rangle$ is 1, (PI) and (IM) act on $|b\rangle$. When Q is the minimum even integer that is $(2^F/(n - 1))^{1/2} = (2^4/9)^{1/2} \approx 1.333 \leq 2 = Q$, the total number that (PI) and (IM) act on $|b\rangle$ is $Q \approx 2$. Next, (OB) observes $|b\rangle$. These actions are repeated sequentially from $|a_1\rangle$ to $|a_9\rangle$. Therefore, each state of $|a_h\rangle$ is 0, 1, \dots , 7, and 8, and the total states become $(n - 1)^{n-1} = 9^9$.

Step 4: (B) changes $|c_0\rangle, |c_1\rangle, \dots, |c_6\rangle$, and $|c_7\rangle$ for $|c_0 + 1\rangle, |c_1 + 1\rangle, \dots, |c_6 + 1\rangle$, and $|c_7 + 1\rangle$ in $a_h = 0, 1, \dots, 6$, and 7, respectively. This action repeats from $|a_1\rangle$ to $|a_9\rangle$. As the target state for $|c_0\rangle$ is 1, (PI) and (IM) act on $|c_0\rangle$. When R_0 is the minimum even integer that is $(9/8)^{8/2} \approx 1.602 \leq 2 = R_0$, the total number that (PI) and (IM) act on $|c_0\rangle$ is R_0 . Next, (OB) observes $|c_0\rangle$. Therefore, only the paths that contain 1 piece of 0 remain. The number of data is $9 \cdot 8^8$. As the target state for $|c_1\rangle$ is 1, (PI) and (IM) act on $|c_1\rangle$. When R_1 is the minimum even integer that is $(8/7)^{7/2} \approx 1.596 \leq 2 = R_1$, the total number that (PI) and (IM) act on $|c_1\rangle$ is R_1 . Next, (OB) observes $|c_1\rangle$. Therefore, only the paths that contain 1 piece of 1 remain. The number of data is $9 \cdot 8 \cdot 7^7$. Similarly, these actions are repeated sequentially from $|c_2\rangle$ to $|c_7\rangle$. Only the paths that contain 1 piece of number from 0 to 8, respectively, remain. The number of data is $9!$ [= W_0].

Step 5: (C_1) changes $|d_1\rangle$ and $|d_2\rangle$ for $|L(0, 1 + a_1) + L(1 + a_1, 1 + a_2)\rangle$ and $|9^8 a_1 + 9^7 a_2\rangle$, respectively, from $|a_1\rangle$ and $|a_2\rangle$. Similarly, (C_i) [$2 \leq i \leq 7$. i is an integer.] changes $|d_1\rangle$ and $|d_2\rangle$ for $|d_1 + L(1 + a_i, 1 + a_{i+1})\rangle$ and $|d_2 + 9^{10-(i+2)} a_{i+1}\rangle$, respectively, from $|a_i\rangle$ and $|a_{i+1}\rangle$. This action is repeated sequentially from $|a_2\rangle$ to $|a_7\rangle$. (C_8) changes $|d_1\rangle$ and $|d_2\rangle$ for $|d_1 + L(1 + a_8, 1 + a_9) + L(1 + a_9, 0)\rangle$ and $|d_2 + 9^0 a_9\rangle$, respectively, from $|a_8\rangle$ and

5. DISCUSSION AND SUMMARY

The computational complexity of this quantum algorithm [= S] becomes the following. In the order of the actions by the gates, the number of them is $F(n - 1)$ at \boxed{H} , $n - 1$ at (A) , $Q(n - 1) \approx 2(n - 1)$ at (PI) and (IM) , $n - 1$ at (OB) , $n - 1$ at (B) , $\sum_{i=0 \rightarrow n-3} R_i \approx 2(n - 2)$ at (PI) and (IM) , $(n - 2)$ at (OB) , $2(n - 2)$ at (C_i) [$0 \leq i \leq n - 2$. i is an integer.], 2 at (D) , g at (E_i) [$1 \leq i \leq g$. i is an integer.], $\sum_{i=1 \rightarrow g} T_i \approx 2g$ at (PI) and (IM) , and g at (OB) . These processes repeated about $\log_2(n - 1)!$. Therefore, S becomes $((F + 10)(n - 1) - 3 + 4g) \log_2(n - 1)!$. In the example of the numerical computation, S is 978. The computational complexity of the classical computation [= Z] is $(n - 1)!/2 = 9!/2 = 181440$. After all, S/Z becomes about $1/186$. When n is large enough, S becomes about $3(\log_2(n - 1))^2(n - 1)^2$, where F is about $\log_2(n - 1)$, g is about $((n - 1)/2)\log_2(n - 1)$ and $n!$ is about $n^n e^{-n} (2\pi n)^{1/2}$ [Stirling's formula], and S/Z is about $3(\log_2(n - 1))^2(n - 1)^2 / ((n - 1)!/2)$. For example, as for $n = 50$, S/Z is about $1/10^{57}$. Therefore, a decreased process becomes possible.

I hope that this result will be confirmed by many experiments.

APPENDIX-1

$U[X]$ [X is the number of datum.] is the number of the repeated permutation of $(n - 1)^{n-1}$ type, and $V(Y)$ [Y is the number of datum.] is the number of permutation of $(n - 1)!$ type.

Therefore, in $(n - 1)!$ type, the numbers of permutation from first to $((n - 1)!/4) - 2$ are, in $(n - 1)^{n-1}$ type, the numbers of permutation from $U(V(Y = 1))$ to $U(V(Y = ((n - 1)!/4) - 2))$, because a number of combinations of answers is at least 2. The order of numbers of permutation from 1 to about $(n - 1)!/4$ converged the probability amplitudes by the Appendix-2.

And then, this process is repeated.

Where, the examples of $U[X] = U(V(Y))$ are shown at the section 4 [Numerical Computation].

Example-1:

$$Y = 1 = (v_1 = 0)8! + (v_2 = 0)7! + (v_3 = 0)6! + (v_4 = 0)5! + (v_5 = 0)4! + (v_6 = 0)3! + (v_7 = 0)2! + (v_8 = 1)1!$$

Therefore, 0, 1, 2, 3, 4, 5, 6, 7, 8 \Rightarrow from (I), $v_1 = v_2 = v_3 = v_4 = v_5 = v_6 = v_7 = 0 \rightarrow t_1 = 0, t_2 = 1, t_3 = 2, t_4 = 3, t_5 = 4, t_6 = 5, t_7 = 6$; 7, 8 \Rightarrow from (IV), $v_8 = 1 \rightarrow t_8 = 7, t_9 = 8$.

And then, $U(V(Y = 1)) = (t_1 = 0)9^8 + (t_2 = 1)9^7 + (t_3 = 2)9^6 + (t_4 = 3)9^5 + (t_5 = 4)9^4 + (t_6 = 5)9^3 + (t_7 = 6)9^2 + (t_8 = 7)9^1 + (t_9 = 8)9^0 = 6053444$.

Example-2:

$Y = (9!/4) - 2 = 90718 = (v_1 = 2)8! + (v_2 = 1)7! + (v_3 = 6)6! + (v_4 = 5)5! + (v_5 = 4)4! + (v_6 = 3)3! + (v_7 = 2)2! + (v_8 = 0)1!$.

Therefore, 0, 1, 2, 3, 4, 5, 6, 7, 8 \Rightarrow from (III), $v_1 = 2 \rightarrow t_1 = 2$; 0, 1, 3, 4, 5, 6, 7, 8 \Rightarrow from (III), $v_2 = 1 \rightarrow t_2 = 1$; 0, 3, 4, 5, 6, 7, 8 \Rightarrow from (III), $v_3 = 6 \rightarrow t_3 = 8$; 0, 3, 4, 5, 6, 7 \Rightarrow from (III), $v_4 = 5 \rightarrow t_4 = 7$; 0, 3, 4, 5, 6 \Rightarrow from (III), $v_5 = 4 \rightarrow t_5 = 6$; 0, 3, 4, 5 \Rightarrow from (III), $v_6 = 3 \rightarrow t_6 = 5$; 0, 3, 4 \Rightarrow from (II), $v_7 = 2, v_8 = 0 \rightarrow t_7 = 3, t_8 = 4, t_9 = 0$.

And then, $U(V(Y = (9!/4) - 2 = 90718)) = (t_1 = 2)9^8 + (t_2 = 1)9^7 + (t_3 = 8)9^6 + (t_4 = 7)9^5 + (t_5 = 6)9^4 + (t_6 = 5)9^3 + (t_7 = 3)9^2 + (t_8 = 4)9^1 + (t_9 = 0)9^0 = 95584572$.

APPENDIX-2

It is assumed that the number of data is N , the value of data of $N/4$ is G , and values of data of $3N/4$ are the others. When the probability amplitudes of data of G are marked a minus, the mean of probability amplitudes becomes $(N^{1/2}(3N/4) - N^{1/2}(N/4))/N = (1/2)N^{1/2}$.

When the inversion about mean is practiced, the probability amplitudes of data of G are

$-(-N^{1/2}) + (1/2)N^{1/2} \times 2 = 2N^{1/2}$, and the sum of square of probability amplitude is $(2N^{1/2})^2(1/4)N = 1$. Therefore, the probability amplitudes of data of others are 0.

After all, the data of $N/4$ of G remain [3, 6, 7, 10].

When this process is repeated, the number of data decreases and the probability amplitudes of necessary data increase.

REFERENCES

- [1] Susa Y., Yamashiro Y., Yamamoto M., and Nishimori H., 2018, “Exponential speedup of quantum annealing by inhomogeneous driving of the transverse field,” *J. Phys. Soc. Jpn.*, 87, 023002.
- [2] Deutsch D., and Jozsa R., 1992, “Rapid solution of problems by quantum computation,” *Proc. Roy. Soc. Lond. A*, 439:553-558.
- [3] Takeuchi S., 2005, *Ryoshi Konpyuta (Quantum Computer)*, Kodansha, Tokyo, Japan [in Japanese].
- [4] Miyano K., and Furusawa A., 2008, *Ryoshi Konpyuta Nyumon (An Introduction to Quantum Computation)*, Nipponhyoronsha, Tokyo, Japan [in Japanese].
- [5] Shor P. W., 1994, “Algorithms for quantum computation: discrete logarithms and factoring,” *Proc. 35th Annu. Symp. Foundations of Computer Science*, IEEE, pp.124-134.
- [6] Grover L. K., 1996, “A fast quantum mechanical algorithm for database search,” *Proc. 28th Annu. ACM Symp. Theory of Computing*, pp.212-219.
- [7] Grover L. K., 1998, “A framework for fast quantum mechanical algorithms,” *Proc. 30th Annu. ACM Symp. Theory of Computing*, pp.53-62.
- [8] Ambainis A., 2008, “Quantum walks and their algorithmic applications,” [http:// arXiv. org/ quant-ph/ arXiv: quant-ph/ 0403120v3](http://arXiv.org/quant-ph/arXiv:quant-ph/0403120v3).
- [9] Bennett C. H., Bernstein E., Brassard G., and Vazirani U., 1996, “Strengths and weaknesses of quantum computing,” [http:// arXiv. org/ quant-ph/ arXiv: quant-ph/ 9701001v1](http://arXiv.org/quant-ph/arXiv:quant-ph/9701001v1).
- [10] Fujimura T., 2014, “Quantum algorithm for 3-SAT problem by numbering method,” *Glob. J. Pure Appl. Math.*, 10:325-330.
- [11] Crescenzi P., and Kann V., Eds., 2005, “A compendium of NP optimization problems,” [http:// www. csc. kth. se/ ~viggo/ wwwcompendium/](http://www.csc.kth.se/~viggo/wwwcompendium/).
- [12] Durr C., and Hoyer P., 1996, “A quantum algorithm for finding the minimum,” [http:// arXiv. org/ quant-ph/ arXiv: quant-ph/ 9607014v2](http://arXiv.org/quant-ph/arXiv:quant-ph/9607014v2).