Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 18, Number 2 (2022), pp. 687-692

http://www.ripublication.com/gjpam.htm

Quantum Algorithm for Traveling Salesman Problem by Matrix Computation with *Y*-Axis-Rotation (-90 degrees)

Toru Fujimura

Art and Physical Education 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 traveling salesman problem by the matrix computation with *y*-axis- rotation (-90 degrees), and its example are reported. For example, in just one-marked-term of four data, the first Grover-iteration becomes 100% probability by the usual Grover method with *z*-axis-rotation (180 degrees) of one-marked-term. However, when *n* is number of work registers for the traveling salesman problem, only one time matrix computation with *y*-axis- rotation (-90 degrees) gets 100% probability of one-marked-term made by a gate.

Keywords: Quantum algorithm, traveling salesman problem, matrix computation, *y*-axis-rotation (-90 degrees), 100% probability.

AMS subject classification: Primary 81-08; Secondary 81-10, 68Q12.

1. Introduction

The traveling salesman problem was discussed by Takeuchi for the complexity. [1] The quantum algorithm for the traveling salesman problem was reported by Fujimura with usual Grover method. [2] Its example was used by many times of Grover-iteration. [1-5]

When I modulated the Grover method for the knapsack problem, and it was assumed that n was number of address qubits, the one time modulated-Grover-iteration with y-axis-rotation (-90 degrees) of one-marked-term made by a gate got 100% probability. [6]

Therefore, because the quantum algorithm for the traveling salesman problem is examined by the matrix computation with y-axis-rotation (-90 degrees) this time, its result is reported.

2. Traveling Salesman Problem

It is the traveling salesman problem to decide a route that turns round n points requests in the shortest distance. A computational complexity of a classical computation is $(n - 1)!/2$ times because a starting point is fixed and counter routes are excluded. [1, 2, 7]

3. Quantum Algorithm

It is assumed that n points of $P_0(x_0, y_0)$, $P_1(x_1, y_1)$, $P_2(x_2, y_2)$, … , $P_{n-2}(x_{n-2}, y_{n-2})$, and $P_{n-1}(x_{n-1}, y_{n-1})$ are set [x_i and y_i are the two dimensional coordinates. $0 \le i \le n - 1$. 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, routes of P_1, P_2, … , P_{n-2}, and P_{n-1} are considered.

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

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

First of all, query quantum registers (= query registers) $|q_i\rangle$ (i is an integer.), marker quantum registers (= marker registers) $|m\rangle$, work quantum registers (= work registers) a_j ($1 \le j \le n - 1$, j is an integer.) (where, a_j contains qubits $|w_{j,u}\rangle$ (u is an necessary qubits number.), QRAM quantum registers (= QRAM registers) $|Qr_s\rangle$ (s is from 1 to necessary number, and an integer.), and address quantum registers (= address registers) $|Ad_t\rangle$ (t is from 1 to necessary number, and an integer.) are prepared.

Step 1: Each qubit of $|q_i\rangle$, $|m\rangle$, a_j, $|Qr_s\rangle$, and $|Ad_t\rangle$ is set $|0\rangle$.
Step 2: The Hadamard gate $\boxed{H}$ [1-7] acts on each qubit of a_j and $|Ad_t\rangle$. It changes them for entangled states.
Step 3: The data of a_j and $|Ad_t\rangle$ are introduced to filter computation Fi-1 and $|Qr_s\rangle$, respectively, where Fi-1 is $1 \le a_1, a_2, \dots, a_{n-1} \le n-1$, and $a_1 \neq a_2 \neq \dots \neq a_{n-1}$.
Step 4: The data of Fi-1 and $|Qr_s\rangle$ are introduced to function computation Fu and filter computation Fi-2, where Fu is $F = ((x_{a1} - x_0)^2 + (y_{a1} - y_0)^2)^{1/2} + ((x_{a2} - x_{a1})^2 + (y_{a2} - y_{a1})^2)^{1/2} + ((x_{a3} - x_{a2})^2 + (y_{a3} - y_{a2})^2)^{1/2} + \dots + ((x_{an-2} - x_{an-1})^2 + (y_{an-2} - y_{an-1})^2)^{1/2} + ((x_0 - x_{an-1})^2 + (y_0 - y_{an-1})^2)^{1/2}$, and in Fi-2, $F \le M_1$ changes from (1, 0) to (0, 1), and others are from (1, 0) to (1, 0).
Furthermore, there are $P_0(x_0, y_0) = P_0(z_0, z_1)$, $P_1(x_1, y_1) = P_1(z_2, z_3)$, $\dots$, $P_{n-2}(x_{n-2}, y_{n-2}) = P_{n-2}(z_{2(n-2)}, z_{2(n-2)+1})$, and $P_{n-1}(x_{n-1}, y_{n-1}) = P_{n-1}(z_{2(n-1)}, z_{2(n-1)+1})$.
Step 5: In the Gate 1, a one-marked-term is presented by $|q_i\rangle$, (0, 1) is presented by $|m\rangle$, and quantum matrix operators, where this gate is used by the n-SAT problem. [3, 8] The one-marked-term's rotation angle is the theta (= -90 degrees) by y-axis.
Step 6: The modulated-flip is done.
Step 7: The modulated-Grover-iteration is done.
Step 8: Each of $|q_i\rangle$, $|m\rangle$, a_j, $|Qr_s\rangle$, and $|Ad_t\rangle$ is read. When $|m\rangle$ is $|1\rangle$, $M_1 \rightarrow M_2$ ($\approx M_1/2$), another when $|m\rangle$ is $|0\rangle$, $M_1 \rightarrow M_2$ ($\approx 2M_1$), and then Step1 to 8 are repeated. After $\log_2 (n-1)!$ Times [9], the minimum value M_{min} is obtained.

4. Matrix Computation

4-1. Matrix of Rotation by *Y*-Axis (-90 degrees)

In the Bloch sphere, the wave function is $\cos(\theta/2)\ |0\rangle + e^{i\phi} \sin(\theta/2)\ |1\rangle$. [1, 3] Therefore, the rotation by y-axis (-90 degrees) is θ = -90 degrees and ϕ = 0 degree. In this situation, $1|0\rangle + 0|1\rangle$ and $0|0\rangle + 1|1\rangle$ are exchanged $2^{-0.5}|0\rangle - 2^{-0.5}|1\rangle$ and $2^{-0.5}|0\rangle + 2^{-0.5}|1\rangle$, respectively. This exchange is the rotation of 180 degrees by the axis of both θ = - 45 degrees and ϕ = 180 degrees. After all, elements of 2×2 matrix of rotation by y-axis (-90 degrees) Ry_{ij} [i, j : 1 or 2] are $Ry_{11} = Ry_{12} = -Ry_{21} = Ry_{22} = 2^{-0.5}$.
Furthermore, when $\boxed{H}_{ij}$ [i, j : 1 or 2] are $\boxed{H}_{11} = \boxed{H}_{12} = \boxed{H}_{21} = -\boxed{H}_{22} = 2^{-0.5}$, [1-7] and $\sim_{ij}$ [i, j : 1 or 2] are $\sim_{11} = \sim_{22} = 0$, and $\sim_{12} = \sim_{21} = 1$, there is $\boxed{H} \sim = Ry$.

4-2. Used Other Matrixes

i and j are 1 or 2 as the following.

Ry^{-1}_{ij} : Ry^{-1}_{11} = - Ry^{-1}_{12} = Ry^{-1}_{21} = Ry^{-1}_{22} = $2^{-0.5}$, where ~ H = Ry^{-1}.
$Ry^{\sim}_{ij}$: - $Ry^{\sim}_{11}$ = $Ry^{\sim}_{12}$ = $Ry^{\sim}_{21}$ = $Ry^{\sim}_{22}$ = $2^{-0.5}$, where ~ Ry = $Ry^{\sim}$.
E_{ij} : $E_{11} = E_{22} = 1$, and $E_{12} = E_{21} = 0$.
ϕ_{ij} : $\phi_{11} = -\phi_{22} = 1$, and $\phi_{12} = \phi_{21} = 0$. [$\phi_{22} = -1 = e^{i\phi}$ = exp{i180(degrees)}] [3]
Ry^2_{ij} : $Ry^2_{11} = Ry^2_{22} = 0$, and $Ry^2_{12} = -Ry^2_{21} = 1$, where RyRy = Ry^2.

4-3. Example of Matrix Computation

It is assumed that n = 6 points are $P_0(x_0, y_0) = P_0(z_0, z_1) = P_0(0, 0)$, $P_1(x_1, y_1) = P_1(z_2, z_3) = P_1(10, 20)$, $P_2(x_2, y_2) = P_2(z_4, z_5) = P_2(10, 0)$, $P_3(x_3, y_3) = P_3(z_6, z_7) = P_3(0, 10)$, $P_4(x_4, y_4) = P_4(z_8, z_9) = P_4(0, 20)$, $P_5(x_5, y_5) = P_5(z_{10}, z_{11}) = P_5(10, 10)$, and there is a distance P_i and P_j : $L(i, j) = L(j, i)$, $F_{\min} = 60 = M_{\min}$, $L(0, 1) = L(2, 4) = 500^{1/2} \approx 22$, $L(0, 2) = L(2, 5) = L(5, 1) = L(1, 4) = L(4, 3) = L(3, 0) = L(3, 5) = 10$, $L(0, 5) = L(2, 3) = L(3, 1) = L(4, 5) = 200^{1/2} \approx 14$, $L(0, 4) = L(1, 2) = 20$, $M_1 = 80$, $M_2 = 40$, $M_3 = 60$ (= $M_{\min}$), $M_4 = 50$, $M_5 = 55$, $M_6 = 58$, $M_7 = 59$, $\log_2(n\text{-}1)! \approx 6.9070 \approx 7$, and theta = -90 degrees by y-axis.

Furthermore, it is assumed that in $|q_i\rangle$, i is $1 \leq i \leq 7$ (an integer), in a_j, j is $1 \leq j \leq 5$ (an integer), in $|Qr_s\rangle$, s is $1 \leq s \leq 5$ (an integer), and in $|Ad_t\rangle$, t is $1 \leq t \leq 4$ (an integer).

An example of matrix computation is the following, where the one-marked-term is 60 = M_3 = $M_{\min}$.

[(1, 0), (1, 0), (0, -1), (0, -1), (0, -1), (0, -1), (1, 0), {(0, -1)}, [(1, 0), (0, 1), (1, 0), {(0, 1), (1, 0), (0, 1)}, (0, 1), (1, 0), (1, 0), {(1, 0), (1, 0), (0, 1)}, (0, 1), (0, 1), (1, 0)], {(1, 0), (0, 1), (1, 0), (0, 1), (1, 0)}, (0, 1), (0, 1), (1, 0), (0, 1)]

//result, [query registers, {marker register}, [work registers], {QRAM registers}, address registers]//

= [Ry, Ry, Ry, Ry, Ry, Ry, Ry, {Ry}, [(1, 0), (0, 1), (1, 0), {(0, 1), (1, 0), (0, 1)}, (0, 1), (1, 0), (1, 0), {(1, 0), (1, 0), (0, 1)}, (0, 1), (0, 1), (1, 0)], {(1, 0), (0, 1), (1, 0), (0, 1), (1, 0)}, (0, 1), (0, 1), (1, 0), (0, 1)]

//modulated-Grover-iteration, {modulated-Grover-iteration}, [work registers], {QRAM registers}, address registers//

× [Ry, Ry, Ry, Ry, Ry, Ry, Ry, {Ry}, [w, -w, w, {-w, w, -w}, -w, w, w, {w, w, -w}, -w, -w, w], {E, E, E, E, E}, E, E, E, E]

//modulated-Flip, {modulated-Flip}, [work registers : w = ($2^{-0.5}$, $2^{-0.5}$), -w = ($2^{-0.5}$, $-2^{-0.5}$)], {QRAM registers}, address registers //

× [~, ~, E, E, E, E, ~, {E}, [(0, 1), (1, 0), (0, 1), {(1, 0), (0, 1), (0, 1)}, (1, 0), (0, 1), (0, 1), {(0, 1), (0, 1), (1, 0)}, (1, 0), (1, 0), (0, 1)], {E, E, E, E, E}, E, E, E, E]

× [Ry, Ry, Ry, Ry, Ry, Ry, Ry, {Ry}, [(1, 0), (1, 0), (1, 0), {(1, 0), (1, 0), (1, 0)}, (1, 0), (1, 0), (1, 0), {(1, 0), (1, 0), (1, 0)}, (1, 0), (1, 0), (1, 0)], {E, E, E, E, E}, E, E, E, E]
× [~, ~, E, E, E, E, ~, {E}, [w, w, w, {w, w, w}, w, w, w, {w, w, w}, w, w, w], {E, E, E, E, E}, E, E, E, E]
//Gate 1, [query registers, {marker register}, [work registers : w = ($2^{-0.5}$, $2^{-0.5}$)], {QRAM registers}, address registers]//
× [q, q, q, q, q, q, q, {m}, [w, w, w, {w, w, w}, w, w, w, {w, w, w}, w, w, w], {Qr, Qr, Qr, Qr, Qr}, Ad, Ad, Ad, Ad]
//Fi-1 → Fu : q = ($2^{-0.5}$, $2^{-0.5}$), {Fu → Fi-2 : m = ($2^{-0.5}$, $2^{-0.5}$)}, [work registers : w = ($2^{-0.5}$, $2^{-0.5}$)], {QRAM registers : Qr = ($2^{-0.5}$, $2^{-0.5}$)}, {address registers : Ad = ($2^{-0.5}$, $2^{-0.5}$)}//
× [E, E, E, E, E, E, E, {E}, [H, H, H, {H, H, H}, H, H, H, {H, H, H}, H, H, H], {E, E, E, E, E}, H, H, H, H]
//query registers, {marker register}, [Hadamard gates], {QRAM registers}, Hadamard gates//
× [(1, 0), (1, 0), (1, 0), (1, 0), (1, 0), (1, 0), (1, 0), {(1, 0)}, [(1, 0), (1, 0), (1, 0), {(1, 0), (1, 0), (1, 0)}, (1, 0), (1, 0), (1, 0), {(1, 0), (1, 0), (1, 0)}, (1, 0), (1, 0), (1, 0)], {(1, 0), (1, 0), (1, 0), (1, 0), (1, 0)}, (1, 0), (1, 0), (1, 0), (1, 0)]
// query registers, {marker register}, [work registers], {QRAM registers}, address registers]//,
where [symbols] is diagonal elements of matrix, and all of others are zero, four of { } are matrix of marker register, work registers a_2, work registers a_4, and QRAM registers, respectively, [numbers] is [(A_1, B_1), (A_2, B_2), (A_3, B_3), (A_4, B_4), (A_5, B_5), (A_6, B_6), (A_7, B_7), {(C, D)}, [(E_1, F_1), (E_2, F_2), (E_3, F_3), {(E_4, F_4), (E_5, F_5), (E_6, F_6)}, (E_7, F_7), (E_8, F_8), (E_9, F_9), {(E_{10}, F_{10}), (E_{11}, F_{11}), (E_{12}, F_{12})}, (E_{13}, F_{13}), (E_{14}, F_{14}), (E_{15}, F_{15})], {(G_1, H_1), (G_2, H_2), (G_3, H_3), (G_4, H_4), (G_5, H_5)}, (K_1, P_1), (K_2, P_2), (K_3, P_3), (K_4, P_4)], $A_r|0\rangle + B_r|1\rangle$, $|A_r|^2 + |B_r|^2 = 1$, $r = 1 \sim 7$ (r is an integer.), $C|0\rangle + D|1\rangle$, $|C|^2 + |D|^2 = 1$, $E_v|0\rangle + F_v|1\rangle$, $|E_v|^2 + |F_v|^2 = 1$, $v = 1 \sim 15$ (v is an integer.), $G_w|0\rangle + H_w|1\rangle$, $|G_w|^2 + |H_w|^2 = 1$, $w = 1 \sim 5$ (w is an integer.), $K_x|0\rangle + P_x|1\rangle$, $|K_x|^2 + |P_x|^2 = 1$, $x = 1 \sim 4$ (x is an integer.), one set of QRAM is values from z_0 to z_{11} {from 00000 to 11111 (QRAM registers : $|Qr_5\rangle$ $|Qr_4\rangle$ $|Qr_3\rangle$ $|Qr_2\rangle$ $|Qr_1\rangle$)}, one set of Ad is values from 0 to 11 in z {from 0000 to 1111 (address registers : $|Ad_4\rangle$ $|Ad_3\rangle$ $|Ad_2\rangle$ $|Ad_1\rangle$), and Fu is F from 60 to 100 {from 0000000 to 1111111 (query registers : $|q_7\rangle$ $|q_6\rangle$ $|q_5\rangle$ $|q_4\rangle$ $|q_3\rangle$ $|q_2\rangle$ $|q_1\rangle$)}.
After all, $P_0 \to P_2 \to P_5 \to P_1 \to P_4 \to P_3 \to P_0$ (or $P_0 \to P_3 \to P_4 \to P_1 \to P_5 \to P_2 \to P_0$) {$L_{\min} = 60$}is an answer.

5. Discussion and Summary

In the matrix computation for the traveling salesman problem, *y*-axis-rotation (-90 degrees) method becomes 100% probability of one-marked-term made by a gate on the one time modulated- Grover-iteration. For getting the minimum value, there are $\log_2 (n - 1)!$ times. In the section 4, there are $\log_2 (6 - 1)! = 7$ times.

I will apply this method for other problems.

References

[1] Takeuchi, S., 2005, Ryoshi Konpyuta (Quantum Computer), Kodansha, Tokyo, Japan [in Japanese].

[2] Fujimura, T., 2012, "Quantum algorithm for traveling salesman problem," Glob. J. Pure Appl. Math., **8**, 65-69.

[3] Johnston, E.R., Harrigan, N., and Gimeno-Segovia, M., 2019, Programing Quantum Computers, O'Reilly, ISBN 978-1-492-03968-6.

[4] Grover, L. K., 1996, "A fast quantum mechanical algorithm for database search," Proc. 28th Annu. ACM Symp. Theory of Computing, pp.212-219.

[5] Grover, L. K., 1998, "A framework for fast quantum mechanical algorithms," Proc. 30th Annu. ACM Symp. Theory of Computing, pp.53-62.

[6] Fujimura, T., 2022, "Quantum algorithm for knapsack problem by matrix computation with *y*-axis-rotation (-90 degrees)," Glob. J. Pure Appl. Math., **18**, 511-516.

[7] Miyano, K., and Furusawa, A., 2008, Ryoshi Konpyuta Nyumon (An Introduction to Quantum Computation), Nipponhyoronsha, Tokyo, Japan [in Japanese].

[8] Cook, S. A., 1971, "The complexity of theorem proving procedures," Proc. 3rd Annu. ACM Symp. Theory of Computing, pp.151-158.

[9] Durr, C., and Hoyer, P., 1996, "A quantum algorithm for finding the minimum," http:// arXiv. org/ quant-qh/ arXiv: quant-ph/ 9607014v2.