Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 18, Number 2 (2022), pp. 511-516

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

# Quantum Algorithm for Knapsack 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 knapsack 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 address qubits for the knapsack 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, knapsack problem, matrix computation, *y*-axis-rotation (-90 degrees), 100% probability.

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

## 1. INTRODUCTION

The knapsack problem was discussed by Takeuchi for the complexity. [1] The quantum algorithm for the knapsack 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 *n*-SAT problem, and it was assumed that *n* was number of qubits, the one time modulated-Grover-iteration with *y*-axis-rotation (-90 degrees) of one-marked-term made by several gates got 100% probability. [6]

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

## 2. KNAPSACK PROBLEM

As for $n$ pieces of different weight luggage, the knapsack problem requests the best combination of the luggage packed into the knapsack that a weight $k$ is assumed to be an upper bound. [1, 2]

When weights of the $n$ pieces of luggage are assumed $x_1$, $x_2$, … , $x_n$, and coefficients in which 0 or 1 are taken are $m_1$, $m_2$, … , $m_n$, a sum of weights becomes $m_1x_1 + m_2x_2 + \ldots + m_nx_n$.

It can be said from the above-mentioned fact the knapsack problem is a problem of requesting the best combination of 0 and 1 of $m_1$, $m_2$, … , $m_n$ in the upper bound weight $k$. [2]

## 3. QUANTUM ALGORITHM

When $n$ is number of address qubits without limit, the matrix computation with $y$-axis-rotation (-90 degrees) gets 100% probability of one-marked-term made by a gate. [6] This gate is used by the $n$-SAT problem. [3, 7]

First of all, query quantum registers (= query registers) $|q_i\rangle$ [$1 \leq i \leq t$. $i$ and $t$ are integers. $t$ is a necessary number for weight.] and address quantum registers (= address registers) $|x_j\rangle$ [$1 \leq j \leq n$. $j$ and $n$ are integers without limit.] are prepared.

Step 1: Each qubit of $|q_i\rangle$ and $|x_j\rangle$ is set $|0\rangle$.

Step 2: The Hadamard gate $\boxed{H}$ [1-6, 8] acts on each qubit of $|x_j\rangle$. It changes them for entangled states.

Step 3: Address data are introduced to QRAM [3], and then weight data [$m_j$ : $j = 1 \rightarrow n$. $j$ is an integer.] are taken out.

Step 4: In a function device, $F = \Sigma_{j=1 \rightarrow n} m_j x_j$ is computed, where $m_j$ is weight.

Step 5: In the Gate 1, a one-marked-term is presented by $|q_i\rangle$, and quantum matrix operators. 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 $|q_i\rangle$ and $|x_j\rangle$ is read. The one-marked-term and address value are 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› + $e^{i\phi}$ sin($\theta$/2) |1›. [1, 3] Therefore, the rotation by *y*-axis (-90 degrees) is $\theta$ = -90 degrees and $\phi$ = 0 degree. In this situation, 1|0› + 0|1› and 0|0› + 1|1› are exchanged $2^{-0.5}$|0› - $2^{-0.5}$|1› and $2^{-0.5}$|0› + $2^{-0.5}$|1›, respectively. This exchange is the rotation of 180 degrees by the axis of both $\theta$ = - 45 degrees and $\phi$ = 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 $\boxed{H}_{11}$ = $\boxed{H}_{12}$ = $\boxed{H}_{21}$ = - $\boxed{H}_{22}$ = $2^{-0.5}$, [1-6, 8] and $\sim_{ij}$ [*i*,*j* : 1 or 2] are $\sim_{11}$ = $\sim_{22}$ = 0, and $\sim_{12}$ = $\sim_{21}$ = 1, there is $\boxed{H}$ ~ = 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 ~ $\boxed{H}$ = $Ry^{-1}$.

$Ry^{\sim}_{ij}$ : - $Ry^{\sim}_{11}$ = $Ry^{\sim}_{12}$ = $Ry^{\sim}_{21}$ = $Ry^{\sim}_{22}$ = $2^{-0.5}$, where ~ Ry = $Ry^{\sim}$.

$E_{ij}$ : $E_{11}$ = $E_{22}$ = 1, and $E_{12}$ = $E_{21}$ = 0.

$\phi_{ij}$ : $\phi_{11}$ = - $\phi_{22}$ = 1, and $\phi_{12}$ = $\phi_{21}$ = 0. [$\phi_{22}$ = -1 = $e^{i\phi}$ = exp{*i*180(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 6 (= *n*) pieces of luggage of weight are $m_1$ = 13kg, $m_2$ = 8kg, $m_3$ = 3kg, $m_4$ = 6kg, $m_5$ = 15kg, $m_6$ = 2kg, and the upper bound of the weight of the knapsack is *k* = 20kg. Furthermore, it is assumed that the one-marked-term = 20 (= *k*), the theta = -90 degrees by *y*-axis, and query register qubits *t* = 6.

An example of matrix computation is the following.

[(0, 1), (1, 0), (0, -1), (1, 0), (0, -1), (1, 0), {(1, 0), (1, 0), (0, -1), (1, 0), (0, -1), (0, -1)}] //result, [query registers, {address registers}]//

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

//modulated-Grover-iteration,{address registers}//

× [Ry, Ry, Ry, Ry, Ry, Ry, { Ap, Ap, -Ap, Ap, -Ap, -Ap}]

//modulated-Flip, {address registers : Ap = ($2^{-0.5}$, $2^{-0.5}$), -Ap = ($2^{-0.5}$, $-2^{-0.5}$)}//

× [~, ~, E, ~, E, ~, {Ae, Ae, Ae, Ae, Ae, Ae}]

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

× [~, ~, E, ~, E, ~, {E, E, E, E, E, E}]

//Gate 1, {Ae = (1, 0), because there is no address for value of the gate 1.}//

× [Fu, Fu, Fu, Fu, Fu, Fu, {E, E, E, E, E, E}]
//Function : Fu = ($2^{-0.5}$, $2^{-0.5}$), {address registers}//
× [Qr, Qr, Qr, Qr, Qr, Qr, {Ad, Ad, Ad, Ad, Ad, Ad}]
//QRAM registers : Qr = ($2^{-0.5}$, $2^{-0.5}$), {address registers : Ad = ($2^{-0.5}$, $2^{-0.5}$)}//
× [E, E, E, E, E, E, {H, H, H, H, H, H}]
//query 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)}]
//query registers, {address registers}//,
where [symbols] is diagonal elements of matrix, and all of others are zero, { } is matrix of address qubits, [numbers] is [($a_1$, $b_1$), ($a_2$, $b_2$), ($a_3$, $b_3$), ($a_4$, $b_4$), ($a_5$, $b_5$), ($a_6$, $b_6$), {($c_1$, $d_1$), ($c_2$, $d_2$), ($c_3$, $d_3$), ($c_4$, $d_4$), ($c_5$, $d_5$), ($c_6$, $d_6$)}], $a_p|0\rangle + b_p|1\rangle$, $|a_p|^2 + |b_p|^2 = 1$, $p = 1 \sim 6\ (= t)$ ($p$ and $t$ are integers.), $c_q|0\rangle + d_q|1\rangle$, $|c_q|^2 + |d_q|^2 = 1$, $q = 1 \sim 6\ (= n)$ ($q$ and $n$ are integers.), Ae is set zero as no value for the gate 1, one set of Ad is values from 000000 to 111111 (address registers : $|x_6\rangle\ |x_5\rangle\ |x_4\rangle\ |x_3\rangle\ |x_2\rangle\ |x_1\rangle$), one set of Qr is $m_1$ : 000001 (address registers : $|x_6\rangle\ |x_5\rangle\ |x_4\rangle\ |x_3\rangle\ |x_2\rangle\ |x_1\rangle$. Same as above.), $m_2$ : 000010, $m_3$ : 000100, $m_4$ : 001000, $m_5$ : 010000, $m_6$ : 100000, $m_{\text{others}}$ : address value (Weight values are zero.), and one set of Fu is $F$ ($\neq$ 0) from 000001 to 111111 (address registers : $|x_6\rangle\ |x_5\rangle\ |x_4\rangle\ |x_3\rangle\ |x_2\rangle\ |x_1\rangle$).

After all, $F$ is $m_1x_1 + m_2x_2 + m_3x_3 + m_4x_4 + m_5x_5 + m_6x_6 = 13\times0 + 8\times0 + 3\times1 + 6\times0 + 15\times1 + 2\times1 = 20$.

## 5. DISCUSSION

In section 4, QRAM is set in query registers. In this section 5, QRAM is set out query registers. Another example of matrix computation is the following, where $x_j$ is an work register, Qr is an QRAM register, and Ad (= $\text{Ad}_s$, $s = 1 \sim 3$) is an address register.

[(0, 1), (1, 0), (0, -1), (1, 0), (0, -1), (1, 0), {(1, 0), (1, 0), (0, -1), (1, 0), (0, -1), (0, -1)}, (0, 1), (0, 1), (0, 1), (0, 1), {(0, 1), (0, 1), (1, 0)}]
//result, [query registers, {work registers}, QRAM registers, {address registers}]//
= [Ry, Ry, Ry, Ry, Ry, Ry, {(1, 0), (1, 0), (0, -1), (1, 0), (0, -1), (0, -1)}, E, E, E, E,{E, E, E}]
//modulated-Grover-iteration, {work registers}, QRAM registers, {address registers}//
× [Ry, Ry, Ry, Ry, Ry, Ry, {Ap, Ap, -Ap, Ap, -Ap, -Ap}, E, E, E, E,{E, E, E}]
//modulated-Flip, {address registers : Ap = ($2^{-0.5}$, $2^{-0.5}$), -Ap = ($2^{-0.5}$, $-2^{-0.5}$)}, QRAM registers, {address registers}//

× [~, ~, E, ~, E, ~, {Ae, Ae, Ae, Ae, Ae, Ae}, E, E, E, E,{E, E, E}]
× [Ry, Ry, Ry, Ry, Ry, Ry, {(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, E}, E, E, E, E,{E, E, E}]
//Gate 1, {Ae = (1, 0), because there is no address for value of the gate 1.}, QRAM registers, {address registers}//
× [Fu, Fu, Fu, Fu, Fu, Fu, {E, E, E, E, E, E} Qr, Qr, Qr, Qr,{Ad, Ad, Ad}]
//Function : Fu = ($2^{-0.5}$, $2^{-0.5}$), {work registers}, QRAM registers: Qr = ($2^{-0.5}$, $2^{-0.5}$), {address registers : Ad = ($2^{-0.5}$, $2^{-0.5}$)}//
× [E, E, E, E, E, E, {$\boxed{\text{H}}$, $\boxed{\text{H}}$, $\boxed{\text{H}}$, $\boxed{\text{H}}$, $\boxed{\text{H}}$, $\boxed{\text{H}}$}, E, E, E, E, {$\boxed{\text{H}}$, $\boxed{\text{H}}$, $\boxed{\text{H}}$}]
//query registers,{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)}]
//query registers, {work registers}, QRAM registers, {address registers}//,
where [symbols] is diagonal elements of matrix, and all of others are zero, two of { } are matrix of work registers and address registers, respectively, [numbers] is [($a_1$, $b_1$), ($a_2$, $b_2$), ($a_3$, $b_3$), ($a_4$, $b_4$), ($a_5$, $b_5$), ($a_6$, $b_6$), {($c_1$, $d_1$), ($c_2$, $d_2$), ($c_3$, $d_3$), ($c_4$, $d_4$), ($c_5$, $d_5$), ($c_6$, $d_6$)}, ($e_1$, $f_1$), ($e_2$, $f_2$), ($e_3$, $f_3$), ($e_4$, $f_4$), {($g_1$, $h_1$), ($g_2$, $h_2$), ($g_3$, $h_3$)}], $a_p|0\rangle + b_p|1\rangle$, $|a_p|^2 + |b_p|^2 = 1$, $p = 1 \sim 6$ (= $t$) ($p$ and $t$ are integers.), $c_q|0\rangle + d_q|1\rangle$, $|c_q|^2 + |d_q|^2 = 1$, $q = 1 \sim 6$ (= $n$) ($q$ and $n$ are integers.), $e_r|0\rangle + f_r|1\rangle$, $|e_r|^2 + |f_r|^2 = 1$, $r = 1 \sim 4$ ($r$ is an integer.), $g_s|0\rangle + h_s|1\rangle$, $|g_s|^2 + |h_s|^2 = 1$, $s = 1 \sim 3$ ($s$ is an integer.), Ae is set zero as no value for the gate 1, one set of Ad is values from 000 to 111 (address registers : $|Ad_3\rangle$ $|Ad_2\rangle$ $|Ad_1\rangle$), one set of Qr is $m_1 = 13$, $m_2 = 8$, $m_3 = 3$, $m_4 = 6$, $m_5 = 15$, $m_6 = 2$, and one set of Fu is $F$ ($\neq 0$) from 000001 to 111111 (work registers : $|x_6\rangle$ $|x_5\rangle$ $|x_4\rangle$ $|x_3\rangle$ $|x_2\rangle$ $|x_1\rangle$).

After all, $F$ is $m_1x_1 + m_2x_2 + m_3x_3 + m_4x_4 + m_5x_5 + m_6x_6 = 13\times0 + 8\times0 + 3\times1 + 6\times0 + 15\times1 + 2\times1 = 20$.

## 6. SUMMARY

In the matrix computation for the knapsack 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.

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., 2010, "Quantum algorithm for knapsack problem," Glob. J. Pure Appl. Math., **6**, 263-266.

[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 $n$-SAT problem by matrix computation with $y$-axis-rotation (-90 degrees)," Glob. J. Pure Appl. Math., **18**, 401-408.

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

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