

Quantum Algorithm for 3-SAT Problem of 19, and 20 Variables by Quantum Fourier Transform with Repeat Qubits, and Weight Changes on QCEngine

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
E-mail: tfujimura8@gmail.com*

Abstract

A quantum algorithm for the 3-SAT problem of 19, and 20 variables by the quantum Fourier transform with the repeat qubits, and the weight changes on the QCEngine, and its example are reported. When there are 3 literals with 2 'OR's in each clause, a weight of r -th clause is m_r [m_r is a natural number. $1 \leq r \leq (\text{number of clauses}) = d$.], the r -th clause is $C_{u,r}(x_1, x_2, x_3, \dots, x_n)$ [u is $2^0x_1 + 2^1x_2 + 2^2x_3 + \dots + 2^{n-1}x_n$. $x_1, x_2, x_3, \dots$, and x_n are the variables, and the repeat qubits.], and $S(u)$ is $\sum_{r=1}^d m_r \times C_{u,r}(x_1, x_2, x_3, \dots, x_n)$, $\text{mod}(S(u)_{\max})$ of $S(u)$ [$S(u)_{\max}$ is the maximum value of $S(u)$.] is computed, next, for u , the quantum Fourier transform is done. In this time, there are 19, and 20 variables, the repeat qubits, and changed weights. The complexity of this method is able to be several times.

Keywords: Quantum algorithm, 3-SAT problem, 19, and 20 variables, quantum Fourier transform, repeat qubits, weight changes, QCEngine.

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

1. Introduction

Feynman discussed about the necessity of quantum computer, because the nature runs by quantum. Deutsch, and Jozsa reported the first quantum algorithm for the exponential accelerated computation. [2] Shor discussed the valid quantum algorithm for factorization. [3] Grover reported the fast quantum algorithm for database search. [4, 5] The complexity of the 3-SAT problem had been discussed by Cook. [6] The quantum computer's example of the 3-SAT problem is reported by Johnston, Harrigan, and Gimeno-Segovia with the QCEngine (free on-line quantum computer simulator). [7] Fujimura discussed a quantum algorithm for the 3-SAT problem by the Shor's Fourier transform with the RAM on the QCEngine. [8] Still more, Fujimura

discussed a quantum algorithm for the 3-SAT problem of 17, and 18 variables by the quantum Fourier transform with the repeat qubits, and the weight changes on the QCEngine. [9]

According to my advanced study, when the quantum Fourier transform with the repeat qubits, and the weight changes on the QCEngine for the 3-SAT problem of 19, and 20 variables is used, the complexity of the 3-SAT problem of 19, and 20 variables is able to be several times.

Therefore, the quantum algorithm for the 3-SAT problem of 19, and 20 variables is examined by the quantum Fourier transform with the repeat qubits, and the weight changes on the QCEngine, and its result are reported.

2. 3-SAT Problem

In the 3-SAT problem, it is assumed that (i) each value of n variables becomes “TRUE”, or “FALSE”, “ $\sim$ ” is “NOT”, “V” is “OR”, “&” is “AND”, (ii) “V”, “ $\sim$ ”, and 3 different variables are included in each parentheses (= clause) that are connected by “&”. If a value of logical formula by the literals, and the logical connectives is “TRUE”, it is decided whether there is at least one combination of values of the variables or not. [6-9]

3. Quantum Algorithm

The following conditions are assumed. (I) Each value of variables $x_1, x_2, x_3, \dots$, and x_n becomes “TRUE” [= 1], or “FALSE” [= 0]. “ $\sim$ ” is “NOT”. “V” is “OR”. “&” is “AND”. For example, it is assumed in this algorithm that (1 V 1 V 1), (1 V 1 V 0), and (1 V 0 V 0) become 1, and (0 V 0 V 0) becomes 0. (II) “V”, “ $\sim$ ”, and 3 different variables in $x_1, x_2, x_3, \dots$, and x_n are included in each clause, and then the clauses are connected by “&”. In these conditions, if a value of logical formula by the literals, and the operators is “TRUE”, it is searched whether there is at least one combination of values of the variables or not. It is assumed that n is number of qubits, u is $2^0x_1 + 2^1x_2 + 2^2x_3 + \dots + 2^{n-1}x_n$, a weight of r -th clause is m_r [m_r is a natural number. $1 \leq r \leq$ (number of clauses) = d .], the r -th clause is $C_{u,r}(x_1, x_2, x_3, \dots, x_n)$, $S(u)$ is $\sum_{r=1 \rightarrow d} m_r \times C_{u,r}(x_1, x_2, x_3, \dots, x_n)$, and $S(u)_{\max}$ is [the maximum value of $S(u)$] = k .

First of all, query quantum registers $|x_i\rangle$ [$1 \leq i \leq n$. i is an integer. n is the number of variables in the logical formula, and the repeat qubits.], and work1 quantum registers $|w_{1,j}\rangle$ [$1 \leq j \leq t$. j , and t are integers. t is a necessary number for $S(u)_{\max} \leq \sum_{f=0 \rightarrow t} 2^f$. f is an integer.].

Step 1: The r data are introduced to the RAM [7].

Step 2: Each qubit of $|x_i\rangle$, and $|w_{1,j}\rangle$ is set $|0\rangle$.

Step 3: The Hadamard gate $\boxed{H}$ [3-5, 7-11] acts on each qubit of $|x_i\rangle$. It changes them for entangled states.

Step 4: Each clause is presented by $|x_i\rangle$, $|w_{1,j}\rangle$, add gate, and quantum operators. For $|x_i\rangle$, RAM[$r - 1$] [RAM has m_r data of $r = 1 \rightarrow d$.] is incremented in $|w_{1,j}\rangle$. In a function, $S(u) = \sum_{r=1 \rightarrow d} m_r \times C_{u,r}(x_1, x_2, x_3, \dots, x_n)$ is computed. This operation makes entangled data base. In this case, n is 19, and 20 variables in the logical formula, and the number of repeat qubits.

Step 5: For $|x_i\rangle$, the quantum Fourier transform (= QFT) [3, 7-11] is done.

Step 6: For $|x_i\rangle$, and $|w_{1,j}\rangle$, the probes are done.

Step 7: For $|x_i\rangle$, the read is done.

Step 8: A number of spikes is estimated by the function (<https://oreilly-qc.github.io/?p=12-4> [7]), where the function `estimate_num_spikes (spike, range) [spike: read value, range:  $2^n$ ]` is used.

Step 9: From candidates of the number of spikes, the repeat period P is obtained.

Step 10: From $u = P = 2^0x_1 + 2^1x_2 + 2^2x_3 + \dots + 2^{n-1}x_n$, when there is $S(P)_{\max} [= \sum_{r=1}^d m_r \times C_{P,r} (x_1, x_2, x_3, \dots, x_n) = k]$, it is the answer [one combination of (value of logical formula) = 1].

Step 11: When the probability is 24%, or more than 24%, the next weight changed computation isn't done.

4. Example of Numerical Computation

4-1. 20 Variables

4-1-1. 20 Variables, 0 Repeat Qubits, and RAM = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]

For example of $n = 20$ [20 variables, and 0 repeat qubit], it is assumed that a logical formula : $(x_3 \vee x_4 \vee x_5) \& (\sim x_1 \vee x_2 \vee x_3) \& (\sim x_3 \vee x_4 \vee x_5) \& (x_3 \vee \sim x_4 \vee x_5) \& (\sim x_2 \vee x_3 \vee \sim x_5) \& (\sim x_3 \vee \sim x_4 \vee x_5) \& (\sim x_3 \vee x_4 \vee \sim x_5) \& (x_3 \vee \sim x_4 \vee \sim x_5) \& (\sim x_3 \vee \sim x_4 \vee \sim x_5) \& (x_4 \vee \sim x_5 \vee \sim x_6) \& (\sim x_5 \vee x_6 \vee \sim x_7) \& (x_6 \vee x_7 \vee \sim x_8) \& (x_7 \vee x_8 \vee \sim x_9) \& (x_8 \vee x_9 \vee \sim x_{10}) \& (x_9 \vee x_{10} \vee \sim x_{11}) \& (x_{10} \vee x_{11} \vee \sim x_{12}) \& (x_{11} \vee x_{12} \vee \sim x_{13}) \& (x_{12} \vee x_{13} \vee \sim x_{14}) \& (x_{13} \vee x_{14} \vee \sim x_{15}) \& (x_{14} \vee x_{15} \vee \sim x_{16}) \& (x_{15} \vee x_{16} \vee \sim x_{17}) \& (x_{16} \vee x_{17} \vee \sim x_{18}) \& (x_{17} \vee x_{18} \vee \sim x_{19}) \& (x_{18} \vee x_{19} \vee \sim x_{20})$, each value of $x_{1 \sim 20}$: $x_1 = x_2 = x_3 = x_4 = x_6 = x_7 = x_8 = x_9 = x_{10} = x_{11} = x_{12} = x_{13} = x_{14} = x_{15} = x_{16} = x_{17} = x_{18} = x_{19} = x_{20} = 0$, $x_5 = 1$, $m_1 = 1$, $m_2 = 2$, $m_3 = 3$, $m_4 = 4$, $m_5 = 5$, $m_6 = 6$, $m_7 = 7$, $m_8 = 8$, $m_9 = 9$, $m_{10} = 10$, $m_{11} = 11$, $m_{12} = 12$, $m_{13} = 13$, $m_{14} = 14$, $m_{15} = 15$, $m_{16} = 16$, $m_{17} = 17$, $m_{18} = 18$, $m_{19} = 19$, $m_{20} = 20$, $m_{21} = 21$, $m_{22} = 22$, $m_{23} = 23$, $m_{24} = 24$, $t = 8$, and $k = (24 + 1)24/2 = 300$. It is noticed that the maximum value of $(n + t)$ on the QCEngine is 28.

An example of program on the QCEngine is the following.

```
10 var a = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]; // RAM_a
20 var query_qubits = 20;
30 var work1_qubits = 8;
40 qc.reset(query_qubits + work1_qubits);
50 var query = qint.new(query_qubits, 'query');
60 var work1 = qint.new(work1_qubits, 'work1');
70 qc.label('q'); // set query
80 query.write(0);
90 query.hadamard();
100 qc.label(' ');
110 qc.label('w1'); // set work1
120 work1.write(0);
130 qc.print(' RAM before increment : ' + a + '\n');
140 var query16 = 16;
150 var work1_0 = 0;
```

```
160 qc.label('increment');
170 qc.not(query.bits(0x4)|query.bits(0x8)|query.bits(0x10));
180 work1.add(a[0],query.bits(0x4)|query.bits(0x8)|query.bits(0x10));
190 qc.not(query.bits(0x4)|query.bits(0x8)|query.bits(0x10));
200 qc.not(query.bits(0x2)|query.bits(0x4));
210 work1.add(a[1],query.bits(0x1)|query.bits(0x2)|query.bits(0x4));
220 qc.not(query.bits(0x2)|query.bits(0x4));
230 qc.not(query.bits(0x8)|query.bits(0x10));
240 work1.add(a[2],query.bits(0x4)|query.bits(0x8)|query.bits(0x10));
250 qc.not(query.bits(0x8)|query.bits(0x10));
260 qc.not(query.bits(0x4)|query.bits(0x10));
270 work1.add(a[3],query.bits(0x4)|query.bits(0x8)|query.bits(0x10));
280 qc.not(query.bits(0x4)|query.bits(0x10));
290 qc.not(query.bits(0x4));
300 work1.add(a[4],query.bits(0x2)|query.bits(0x4)|query.bits(0x10));
310 qc.not(query.bits(0x4));
320 qc.not(query.bits(0x10));
330 work1.add(a[5],query.bits(0x4)|query.bits(0x8)|query.bits(0x10));
340 qc.not(query.bits(0x10));
350 qc.not(query.bits(0x8));
360 work1.add(a[6],query.bits(0x4)|query.bits(0x8)|query.bits(0x10));
370 qc.not(query.bits(0x8));
380 qc.not(query.bits(0x4));
390 work1.add(a[7],query.bits(0x4)|query.bits(0x8)|query.bits(0x10));
400 qc.not(query.bits(0x4));
410 work1.add(a[8],query.bits(0x4)|query.bits(0x8)|query.bits(0x10));
420 qc.not(query.bits(0x8));
430 work1.add(a[9],query.bits(0x8)|query.bits(0x10)|query.bits(0x20));
440 qc.not(query.bits(0x8));
450 qc.not(query.bits(0x20));
460 work1.add(a[10],query.bits(0x10)|query.bits(0x20)|query.bits(0x40));
470 qc.not(query.bits(0x20));
480 qc.not(query.bits(0x20)|query.bits(0x40));
490 work1.add(a[11],query.bits(0x20)|query.bits(0x40)|query.bits(0x80));
500 qc.not(query.bits(0x20)|query.bits(0x40));
510 qc.not(query.bits(0x40)|query.bits(0x80));
520 work1.add(a[12],query.bits(0x40)|query.bits(0x80)|query.bits(0x100));
530 qc.not(query.bits(0x40)|query.bits(0x80));
540 qc.not(query.bits(0x80)|query.bits(0x100));
550 work1.add(a[13],query.bits(0x80)|query.bits(0x100)|query.bits(0x200));
560 qc.not(query.bits(0x80)|query.bits(0x100));
570 qc.not(query.bits(0x100)|query.bits(0x200));
580 work1.add(a[14],query.bits(0x100)|query.bits(0x200)|query.bits(0x400));
590 qc.not(query.bits(0x100)|query.bits(0x200));
600 qc.not(query.bits(0x200)|query.bits(0x400));
```

```

610 work1.add(a[15],query.bits(0x200)|query.bits(0x400)|query.bits(0x800));
620 qc.not(query.bits(0x200)|query.bits(0x400));
630 qc.not(query.bits(0x400)|query.bits(0x800));
640 work1.add(a[16],query.bits(0x400)|query.bits(0x800)|query.bits(0x1000));
650 qc.not(query.bits(0x400)|query.bits(0x800));
660 qc.not(query.bits(0x800)|query.bits(0x1000));
670 work1.add(a[17],query.bits(0x800)|query.bits(0x1000)|query.bits(0x2000));
680 qc.not(query.bits(0x800)|query.bits(0x1000));
690 qc.not(query.bits(0x1000)|query.bits(0x2000));
700 work1.add(a[18],query.bits(0x1000)|query.bits(0x2000)|query.bits(0x4000));
710 qc.not(query.bits(0x1000)|query.bits(0x2000));
720 qc.not(query.bits(0x2000)|query.bits(0x4000));
730 work1.add(a[19],query.bits(0x2000)|query.bits(0x4000)|query.bits(0x8000));
740 qc.not(query.bits(0x2000)|query.bits(0x4000));
750 qc.not(query.bits(0x4000)|query.bits(0x8000));
760 work1.add(a[20],query.bits(0x4000)|query.bits(0x8000)|query.bits(0x10000));
770 qc.not(query.bits(0x4000)|query.bits(0x8000));
780 qc.not(query.bits(0x8000)|query.bits(0x10000));
790 work1.add(a[21],query.bits(0x8000)|query.bits(0x10000)|query.bits(0x20000));
800 qc.not(query.bits(0x8000)|query.bits(0x10000));
810 qc.not(query.bits(0x10000)|query.bits(0x20000));
820 work1.add(a[22],query.bits(0x10000)|query.bits(0x20000)|query.bits(0x40000));
830 qc.not(query.bits(0x10000)|query.bits(0x20000));
840 qc.not(query.bits(0x20000)|query.bits(0x40000));
850 work1.add(a[23],query.bits(0x20000)|query.bits(0x40000)|query.bits(0x80000));
860 qc.not(query.bits(0x20000)|query.bits(0x40000));
870 qc.label('QFT');
880 query.QFT();
890 var prob16 = 0;
900 prob16 += query.peekProbability(query16);
910 // Print output query-Prob
920 qc.print(' Prob_query16: ' + prob16);
930 var prob0 = 0;
940 prob0 += work1.peekProbability(work1_0);
950 // Print output work1-Prob
960 qc.print(' Prob_work1_0: ' + prob0);
970 //read
980 qc.label('Rq');
990 var b2 = query.read();
1000 // Print output result
1010 qc.print(' Read query = ' + b2 + '.');
1020 // end

```

When this program is copied on Programming Quantum Computers <https://oreilly-qc.github.io/#> [free on-line quantum computation simulator QCEngine] [7], you can run it. [Caution!: Please delate the line numbers.]

A result of this program is the following.

The probability probe value of $|w_{1,j}\rangle = 0 : \approx 0.00000095367 (\approx 1/1048576)$.

The probability probe value of $|x_i\rangle = 16 : \approx 0.00063633$.

The example of 50 times test : The read value of $|x_i\rangle$; $R_q = 1015856, 163680, 1037312, 97261, 32784, 278544, 845570, 20544, 1007904, 73732, 1007680, 884700, 197632, 917419, 98112, 1047808, 1017276, 651328, 40772, 1047752, 1032192, 1014592, 380896, 952672, 545136, 292600, 1015912, 720888, 135152, 291040, 950792, 2064, 1021958, 926154, 1000512, 343976, 968448, 67520, 719872, 53220, 28648, 1042274, 864, 98600, 905520, 130624, 986762, 3320, 941888, 41990$. (= spike)

The candidates of number of spikes are estimated by the function [the function estimate_num_spikes (spike, range) [spike : read value, range : $2^n = 2^{20} = 1048576$]] : $R_q \rightarrow$ candidates ; $1015856 \rightarrow 32, \dots$; $163680 \rightarrow 6, 13, 19, \dots$; $1037312 \rightarrow 93, \dots$; $97261 \rightarrow 11, 22, \dots$; $32784 \rightarrow 32, \dots$; $278544 \rightarrow 4, 8, 11, 15, 30, \dots$; $845570 \rightarrow 5, 10, 16, \dots$; $20544 \rightarrow 51, \dots$; $1007904 \rightarrow 26, \dots$; $73732 \rightarrow 14, 28, \dots$; $1007680 \rightarrow 26, \dots$; $884700 \rightarrow 6, 13, 19, \dots$; $197632 \rightarrow 5, 11, 16, \dots$; $917419 \rightarrow 8, 16, \dots$; $98112 \rightarrow 11, 21, \dots$; $1047808 \rightarrow 1365, \dots$; $1017276 \rightarrow 34, \dots$; $651328 \rightarrow 3, 5, 8, 16, \dots$; $40772 \rightarrow 26, \dots$; $1047752 \rightarrow 1273, \dots$; $1032192 \rightarrow 64, \dots$; $1014592 \rightarrow 31, \dots$; $380896 \rightarrow 3, 6, 8, 11, 22, \dots$; $952672 \rightarrow 11, 22, \dots$; $545136 \rightarrow 2, 4, 6, 8, 10, 13, 15, 17, \dots$; $292600 \rightarrow 4, 7, 11, 18, \dots$; $1015912 \rightarrow 32, \dots$; $720888 \rightarrow 3, 6, 10, 13, 16, \dots$; $135152 \rightarrow 8, 16, \dots$; $291040 \rightarrow 4, 7, 11, 18, \dots$; $950792 \rightarrow 11, 21, \dots$; $2064 \rightarrow 508, \dots$; $1021958 \rightarrow 39, \dots$; $926154 \rightarrow 9, 17, \dots$; $1000512 \rightarrow 22, \dots$; $343976 \rightarrow 3, 6, 9, 12, 15, 18, \dots$; $968448 \rightarrow 13, 26, \dots$; $67520 \rightarrow 16, \dots$; $719872 \rightarrow 3, 6, 10, 13, 16, \dots$; $53220 \rightarrow 20, \dots$; $28648 \rightarrow 37, \dots$; $1042274 \rightarrow 166, \dots$; $864 \rightarrow 1214, \dots$; $98600 \rightarrow 11, 21, \dots$; $905520 \rightarrow 7, 15, 22, \dots$; $130624 \rightarrow 8, 16, \dots$; $986762 \rightarrow 17, \dots$; $3320 \rightarrow 316, \dots$; $941888 \rightarrow 10, 20, \dots$; $41990 \rightarrow 25, \dots$.

When u is 16 ($2^0x_1 + 2^1x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5 + 2^5x_6 + 2^6x_7 + 2^7x_8 + 2^8x_9 + 2^9x_{10} + 2^{10}x_{11} + 2^{11}x_{12} + 2^{12}x_{13} + 2^{13}x_{14} + 2^{14}x_{15} + 2^{15}x_{16} + 2^{16}x_{17} + 2^{17}x_{18} + 2^{18}x_{19} + 2^{19}x_{20} = 2^0 \times 0 + 2^1 \times 0 + 2^2 \times 0 + 2^3 \times 0 + 2^4 \times 1 + 2^5 \times 0 + 2^6 \times 0 + 2^7 \times 0 + 2^8 \times 0 + 2^9 \times 0 + 2^{10} \times 0 + 2^{11} \times 0 + 2^{12} \times 0 + 2^{13} \times 0 + 2^{14} \times 0 + 2^{15} \times 0 + 2^{16} \times 0 + 2^{17} \times 0 + 2^{18} \times 0 + 2^{19} \times 0 = 16$), the value of logical formula is 1. Therefore, it is the answer.

4 - 2.19 Variables

4-2-1. 19 Variables, 1 Repeat Qubit, and RAM = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23]

For example of $n = 20$ [19 variables, and 1 repeat qubit], it is assumed that a logical formula : $(x_3 \vee x_4 \vee x_5) \& (\sim x_1 \vee x_2 \vee x_3) \& (\sim x_3 \vee x_4 \vee x_5) \& (x_3 \vee \sim x_4 \vee x_5) \& (\sim x_2 \vee x_3 \vee \sim x_5) \& (\sim x_3 \vee \sim x_4 \vee x_5) \& (\sim x_3 \vee x_4 \vee \sim x_5) \& (x_3 \vee \sim x_4 \vee \sim x_5) \& (\sim x_3 \vee \sim x_4 \vee \sim x_5) \& (x_4 \vee \sim x_5 \vee \sim x_6) \& (\sim x_5 \vee x_6 \vee \sim x_7) \& (x_6 \vee x_7 \vee \sim x_8) \& (x_7 \vee x_8 \vee \sim x_9) \& (x_8 \vee x_9 \vee \sim x_{10}) \& (x_9 \vee x_{10} \vee \sim x_{11}) \& (x_{10} \vee x_{11} \vee \sim x_{12}) \& (x_{11} \vee x_{12} \vee \sim x_{13}) \& (x_{12} \vee x_{13} \vee \sim x_{14}) \& (x_{13} \vee x_{14} \vee \sim x_{15}) \& (x_{14} \vee x_{15} \vee \sim x_{16}) \& (x_{15} \vee x_{16} \vee \sim x_{17}) \& (x_{16} \vee x_{17} \vee \sim x_{18}) \& (x_{17} \vee x_{18} \vee \sim x_{19})$, each value of $x_{1 \sim 19}$: $x_1 = x_2 = x_3 = x_4 = x_6 = x_7 = x_8 = x_9 = x_{10} = x_{11} = x_{12} = x_{13} = x_{14} = x_{15} = x_{16} = x_{17} = x_{18} = x_{19} = 0$, $x_5 = 1$, $m_1 = 1$, $m_2 = 2$, $m_3 = 3$, $m_4 = 4$, $m_5 = 5$, $m_6 = 6$, $m_7 = 7$, $m_8 = 8$, $m_9 = 9$, $m_{10} = 10$, $m_{11} = 11$, $m_{12} = 12$, $m_{13} = 13$, m_{14}

$= 14, m_{15} = 15, m_{16} = 16, m_{17} = 17, m_{18} = 18, m_{19} = 19, m_{20} = 20, m_{21} = 21, m_{22} = 22, m_{23} = 23, t = 8, \text{ and } k = (23 + 1)23/2 = 276.$

A result of this problem is the following.

The probability probe value of $|w_{1,j}\rangle = 0 : \approx 0.0000019073 (\approx 1/524288).$

The probability probe value of $|x_i\rangle = 16 : \approx 0.00075283.$

The example of 50 times test : The read value of $|x_i\rangle ; R_q = 63538, 1005520, 1007808, 622592, 188432, 883968, 1023488, 12338, 72836, 387288, 29472, 983032, 1015852, 35328, 24576, 0, 884744, 384948, 656640, 917498, 57408, 139324, 621568, 893906, 901376, 132328, 65152, 918532, 618484, 1015936, 852096, 884608, 919512, 638720, 279276, 758912, 292184, 28672, 1046016, 196598, 933888, 335712, 32756, 901328, 1048528, 614432, 49140, 116144, 1004544, 968. (= spike)$

The candidates of number of spikes are estimated by the function [the function estimate_num_spikes (spike, range) [spike : read value, range : $2^n = 2^{20} = 1048576$]] : $R_q \rightarrow \text{candidates} ; 63538 \rightarrow 17, \dots ; 1005520 \rightarrow 24, \dots ; 1007808 \rightarrow 26, \dots ; 622592 \rightarrow 3, 5, 10, 15, 17, \dots ; 188432 \rightarrow 6, 11, 22, \dots ; 883968 \rightarrow 6, 13, 19, \dots ; 1023488 \rightarrow 42, \dots ; 12338 \rightarrow 85, \dots ; 72836 \rightarrow 14, 29, \dots ; 387288 \rightarrow 3, 5, 8, 16, \dots ; 29472 \rightarrow 36, \dots ; 983032 \rightarrow 16, \dots ; 1015852 \rightarrow 32, \dots ; 35328 \rightarrow 30, \dots ; 24576 \rightarrow 43, \dots ; 0 \rightarrow \text{nothingness} ; 884744 \rightarrow 6, 13, 19, \dots ; 384948 \rightarrow 3, 5, 8, 11, 19, \dots ; 656640 \rightarrow 3, 5, 8, 16, \dots ; 917498 \rightarrow 8, 16, \dots ; 57408 \rightarrow 18, \dots ; 139324 \rightarrow 8, 15, 30, \dots ; 621568 \rightarrow 3, 5, 10, 15, 17, \dots ; 893906 \rightarrow 7, 14, 20, \dots ; 901376 \rightarrow 7, 14, 21, \dots ; 132328 \rightarrow 8, 16, \dots ; 65152 \rightarrow 16, \dots ; 918532 \rightarrow 8, 16, \dots ; 618484 \rightarrow 3, 5, 10, 12, 17, \dots ; 1015936 \rightarrow 32, \dots ; 852096 \rightarrow 5, 11, 16, \dots ; 884608 \rightarrow 6, 13, 19, \dots ; 919512 \rightarrow 8, 16, \dots ; 638720 \rightarrow 3, 5, 10, 13, 18, \dots ; 279276 \rightarrow 4, 8, 11, 15, 30, \dots ; 758912 \rightarrow 4, 7, 11, 18, \dots ; 292184 \rightarrow 4, 7, 11, 18, \dots ; 28672 \rightarrow 37, \dots ; 1046016 \rightarrow 410, \dots ; 196598 \rightarrow 5, 11, 16, \dots ; 933888 \rightarrow 9, 18, \dots ; 335712 \rightarrow 3, 6, 9, 13, 16, \dots ; 32756 \rightarrow 32, \dots ; 901328 \rightarrow 7, 14, 21, \dots ; 1048528 \rightarrow 21845, \dots ; 614432 \rightarrow 3, 5, 10, 12, 17, \dots ; 49140 \rightarrow 21, \dots ; 116144 \rightarrow 9, 18, \dots ; 1004544 \rightarrow 24, \dots ; 968 \rightarrow 1083, \dots .$

When u is 16 ($2^0x_1 + 2^1x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5 + 2^5x_6 + 2^6x_7 + 2^7x_8 + 2^8x_9 + 2^9x_{10} + 2^{10}x_{11} + 2^{11}x_{12} + 2^{12}x_{13} + 2^{13}x_{14} + 2^{14}x_{15} + 2^{15}x_{16} + 2^{16}x_{17} + 2^{17}x_{18} + 2^{18}x_{19} + 2^{19}x_{20} = 2^0 \times 0 + 2^1 \times 0 + 2^2 \times 0 + 2^3 \times 0 + 2^4 \times 1 + 2^5 \times 0 + 2^6 \times 0 + 2^7 \times 0 + 2^8 \times 0 + 2^9 \times 0 + 2^{10} \times 0 + 2^{11} \times 0 + 2^{12} \times 0 + 2^{13} \times 0 + 2^{14} \times 0 + 2^{15} \times 0 + 2^{16} \times 0 + 2^{17} \times 0 + 2^{18} \times 0 + 2^{19} \times 0 = 16$), the value of logical formula is 1. Therefore, it is the answer.

4-2-2.19 Variables, 0 Repeat Qubit, and RAM = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 20, 21, 22, 23]

For example of $n = 19$ [19 variables, and 0 repeat qubit], it is assumed that the logical formula, each value of $x_{1 \sim 19}$, each value of $m_{1 \sim 23}$, t , and k are same in the section 4-2-1.

A result of this problem is the following.

The probability probe value of $|w_{1,j}\rangle = 0 : \approx 0.0000019073 (\approx 1/524288).$

The probability probe value of $|x_i\rangle = 16 : \approx 0.00075283.$

The example of 50 times test : The read value of $|x_i\rangle ; R_q = 257024, 15633, 505218, 354304, 523870, 524280, 194560, 401392, 475792, 33232, 16962, 11858, 32829, 49152, 47648, 508880, 418400, 109683, 502784, 367997, 40832, 476157, 356559,$

491864, 394784, 475920, 437504, 452396, 524287, 508160, 88252, 376208, 31740, 12288, 475183, 336152, 310222, 446207, 500696, 507896, 207736, 377848, 457364, 474336, 36528, 418176, 508928, 307732, 511440, 507776. (= spike)

The candidates of number of spikes are estimated by the function [the function estimate_num_spikes (spike, range) [spike : read value, range : $2^n = 2^{19} = 524288$]] : $R_q \rightarrow$ candidates ; 257024 $\rightarrow$ 2, 4, 6, 8, 10, 12, 14, 16, ... ; 15633 $\rightarrow$ 34, ... ; 505218 $\rightarrow$ 28, ... ; 354304 $\rightarrow$ 3, 6, 9, 12, 15, 19, ... ; 523870 $\rightarrow$ 1254, ... ; 524280 $\rightarrow$ 65536, ... ; 194560 $\rightarrow$ 3, 5, 8, 16, ... ; 401392 $\rightarrow$ 4, 9, 13, 17, ... ; 475792 $\rightarrow$ 11, 22, ... ; 33232 $\rightarrow$ 16, ... ; 16962 $\rightarrow$ 31, ... ; 11858 $\rightarrow$ 44, ... ; 32829 $\rightarrow$ 16, ... ; 49152 $\rightarrow$ 11, 21, ... ; 47648 $\rightarrow$ 11, 22, ... ; 508880 $\rightarrow$ 34, ... ; 418400 $\rightarrow$ 5, 10, 15, 20, ... ; 109683 $\rightarrow$ 5, 10, 14, 19, ... ; 502784 $\rightarrow$ 24, ... ; 367997 $\rightarrow$ 3, 7, 10, 20, ... ; 40832 $\rightarrow$ 13, 26, ... ; 476157 $\rightarrow$ 11, 22, ... ; 356559 $\rightarrow$ 3, 6, 9, 13, 16, ... ; 491864 $\rightarrow$ 16, ... ; 394784 $\rightarrow$ 4, 8, 12, 16, ... ; 475920 $\rightarrow$ 11, 22, ... ; 437504 $\rightarrow$ 6, 12, 18, ... ; 452396 $\rightarrow$ 7, 15, ... ; 524287 $\rightarrow$ nothingness ; 508160 $\rightarrow$ 33, ... ; 88252 $\rightarrow$ 6, 12, 18, ... ; 376208 $\rightarrow$ 4, 7, 14, 21, ... ; 31740 $\rightarrow$ 17, ... ; 12288 $\rightarrow$ 43, ... ; 475183 $\rightarrow$ 11, 21, ... ; 336152 $\rightarrow$ 3, 6, 8, 11, 14, 25, ... ; 310222 $\rightarrow$ 3, 5, 10, 15, 17, ... ; 446207 $\rightarrow$ 7, 13, 20, ... ; 500696 $\rightarrow$ 22, ... ; 507896 $\rightarrow$ 32, ... ; 207736 $\rightarrow$ 3, 5, 10, 15, 20, ... ; 377848 $\rightarrow$ 3, 6, 9, 12, 15, 18, ... ; 492224 $\rightarrow$ 2, 4, 6, 9, 11, 13, 15, 17, ... ; 24 $\rightarrow$ 43691, ... ; 196224 $\rightarrow$ 4, 7, 14, 18, ... ; 457364 $\rightarrow$ 8, 16, ... ; 474336 $\rightarrow$ 11, 21, ... ; 36528 $\rightarrow$ 14, 29, ... ; 418176 $\rightarrow$ 5, 10, 15, 20, ... ; 508928 $\rightarrow$ 34, ... ; 307732 $\rightarrow$ 3, 5, 10, 12, 17, ... ; 511440 $\rightarrow$ 41, ... ; 507776 $\rightarrow$ 32,

When u is 16 ($2^0x_1 + 2^1x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5 + 2^5x_6 + 2^6x_7 + 2^7x_8 + 2^8x_9 + 2^9x_{10} + 2^{10}x_{11} + 2^{11}x_{12} + 2^{12}x_{13} + 2^{13}x_{14} + 2^{14}x_{15} + 2^{15}x_{16} + 2^{16}x_{17} + 2^{17}x_{18} + 2^{18}x_{19} = 2^0 \times 0 + 2^1 \times 0 + 2^2 \times 0 + 2^3 \times 0 + 2^4 \times 1 + 2^5 \times 0 + 2^6 \times 0 + 2^7 \times 0 + 2^8 \times 0 + 2^9 \times 0 + 2^{10} \times 0 + 2^{11} \times 0 + 2^{12} \times 0 + 2^{13} \times 0 + 2^{14} \times 0 + 2^{15} \times 0 + 2^{16} \times 0 + 2^{17} \times 0 + 2^{18} \times 0 = 16$), the value of logical formula is 1. Therefore, it is the answer.

5. Discussion

In the 3-SAT problem, when [(the logical formula) = 1] is obtained, there is only one combination.

5-1. 20 Variables

5-1-1. 20 Variables, 0 Repeat Qubit, and RAM = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]

In the section 4-1, there is RAM = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]. And then, [(the logical formula) = 1] of combination of variables is selected. When N is 2^{20} , in the Grover's method, the complexity is $N^{1/2} = 2^{20/2} = 1024$, in the new method, for (variables, repeat qubits) = (20, 0), it is $50/9 \approx 6$.

In this range, the new method is less than the complexity of the Grover's method, and then, in for (20, 0), the probability is 18%.

5-1-2. 20 Variables, 0 Repeat Qubit, and RAM = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 1]

There are $RAM = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 1]$, $t = 8$, and $k = 277$. And then, [(the logical formula) = 1] of combination of variables is selected. In the new method, for (20, 0), it is $50/4 \approx 13$.

In this range, the new method is less than the complexity of the Grover's method, and then, in for (20, 0), the probability is 8%.

5-1-3. 20 Variables, 0 Repeat Qubit, and $RAM = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 1, 2]$

There are $RAM = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 1, 2]$, $t = 8$, and $k = 256$. And then, [(the logical formula) = 1] of combination of variables is selected. In the new method, for (20, 0), it is $50/10 = 5$.

In this range, the new method is less than the complexity of the Grover's method, and then, in for (20, 0), the probability is 20%.

5-1-4. 20 Variables, Repeat Qubit from 0 to 1, and $RAM = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 1, 2, 3]$

There are $RAM = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 1, 2, 3]$, $t = 7$, and $k = 239$. And then, [(the logical formula) = 1] of combination of variables is selected, and the computation of repeats is done. In the new method, for (20, 0), it is $50/10 = 5$, and for (20, 1), it is $50/13 \approx 4$.

In this range, the new method is less than the complexity of the Grover's method, and then, in for (20, 1), the probability is the maximum value 26%.

From step 11 of section 3 (Quantum Algorithm), the next weight changed computation isn't done.

5-2. 19 Variables

5-2-1. 19 Variables, Repeat Qubit from 0 to 1, and $RAM = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23]$

In the section 4-2, there is $RAM = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23]$. And then, [(the logical formula) = 1] of combination of variables is selected, and the computation of repeat is done. When N is 2^{19} , in the Grover's method, the complexity is $N^{1/2} = 2^{19/2} \approx 724$, in the new method, for (19, 0), it is $50/8 \approx 6$, and for (19, 1), it is $50/11 \approx 5$.

In this range, the new method is less than the complexity of the Grover's method, and then, in for (19, 1), the probability is the maximum value 22%.

5-2-2. 19 Variables, Repeat Qubits from 0 to 2, and $RAM = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 1]$

There are $RAM = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 1]$, $t = 7$, and $k = 254$. And then, [(the logical formula) = 1] of combination of variables is selected, and the computation of repeats is done. In the new method, for (19, 0), it is $50/11 \approx 5$, for (19, 1), it is $50/11 \approx 5$, and for (19, 2), it is $50/5 = 10$.

In this range, the new method is less than the complexity of the Grover's method, and then, in for (19, 0), and (19, 1), the probability is the maximum value 22%.

5-2-3. 19 Variables, Repeat Qubits from 0 to 2, and $RAM = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 1, 2]$

There are $RAM = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 1, 2]$, $t = 7$, and $k = 234$. And then, [(the logical formula) = 1] of combination of

variables is selected, and the computation of repeats is done. In the new method, for (19, 0), it is $50/11 \approx 5$, for (19, 1), it is $50/5 = 10$, and for (19, 2), it is $50/15 \approx 3$.

In this range, the new method is less than the complexity of the Grover's method, and then, in for (19, 2), the probability is the maximum value 30%.

From step 11 of section 3 (Quantum Algorithm), the next weight changed computation isn't done.

6. Summary

The quantum algorithm for the 3-SAT problem of 19, and 20 variables by the quantum Fourier transform with the repeat qubits, and the weight changes on the QCEngine, and its example are reported.

The complexity of this method is several times, and then, in for (19, 2) of $RAM = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 1, 2]$, the probability is the maximum value 30%, and in for (20, 1) of $RAM = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 1, 2, 3]$, the probability is the maximum value 26%.

I will apply this method for other problems.

References

- [1] Feynman, R., 1982, "Simulating physics with computers," *Int. J. Theor. Phys.*, 21, 467-488.
- [2] Deutsch, D., and Jozsa, R., 1992, "Rapid solution of problems by quantum computation," *Proc. Roy. Soc. Lond. A*, 439, 553-558.
- [3] Shor, P. W., 1994, "Algorithms for quantum computation : discrete logarithms and factoring," *Proc. 35th Annu. Symp. Foundations of Computer Science*, IEEE, pp.124-134.
- [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] Cook, S. A., 1971, "The complexity of theorem proving procedures," *Proc. 3rd Annu. ACM Symp. Theory of Computing*, pp.151-158.
- [7] Johnston, E.R., Harrigan, N., and Gimeno-Segovia, M., 2019, *Programming Quantum Computers*, O'Reilly, ISBN 978-1-492-03968-6.
- [8] Fujimura, T., 2024, "Quantum algorithm for 3-SAT problem by Shor's Fourier transform with RAM on QCEngine," *Glob. J. Pure Appl. Math.*, 20, 695-703.
- [9] Fujimura, T., 2026, "Quantum algorithm for 3-SAT problem of 17, and 18 variables by quantum Fourier transform with repeat qubits, and weight changes on QCEngine," *Glob. J. Pure Appl. Math.*, 22, 429-443.

- [10] Takeuchi, S., 2005, Ryoshi Konpyuta (Quantum Computer), Kodansha, Tokyo, Japan [in Japanese].
- [11] Miyano, K., and Furusawa, A., 2008, Ryoshi Konpyuta Nyumon (An Introduction to Quantum Computation), Nipponhyoronsha, Tokyo, Japan [in Japanese]