QUANTUM ANNEALING FOR SUBSET PRODUCT AND NOISY SUBSET PRODUCT

TREY LI

ABSTRACT. In recent works of Li the noisy subset product problem (also known as subset product with errors) was invented and applied to cryptography. To better understand its hardness, we give a quantum annealing algorithm for it. Our algorithm is the first algorithm for the problem. We also give the first quantum annealing algorithm for the subset product problem. The efficiencies of both algorithms rely on the fundamental efficiency of quantum annealing. At the end we give two lattice algorithms for both problems via solving the closest vector problem. The complexities of the lattice algorithms depend on the complexities of solving the closest vector problem in two special lattices. They are efficient when the special closest vector problems fall into the regime of bounded distance decoding problems that can be efficiently solved using existing methods based on the LLL algorithm or Babai's nearest plane algorithm.

1. INTRODUCTION

The subset product problem is a classical NP-complete problem [GJ79, p. 224]. It has not gained much attention for several decades, probably due to the lack of applications. In the recent works [Li22a; Li22d; Li22e; Li22f; Li22g; Li22h; Li22i] Li created the noisy subset product problem and gave applications to cryptography.

To understand the hardness of noisy subset product, Li studied subset product and proposed two algorithms for it, they are the Jacobi symbol parity checking algorithm [Li22b] and the power residue symbol order detecting algorithm [Li22c]. However these algorithms are only reasonable for special cases of the problem. Also, these algorithms cannot be used to solve noisy subset product. In fact, no algorithm for noisy subset product has been discovered. In this paper we give quantum annealing algorithms for the general cases of subset product and noisy subset product.

Quantum annealing was originated from [RCC89] where quantum fluctuations were found to be helpful for finding the lowest energy state of Ising spin glasses; and it is formulated in [KN98] where quantum fluctuations were introduced into simulated annealing.

Before our work, quantum annealing has been used to solve the traveling salesman problem [MST04], the graph coloring problem [TC11a; TC11b; TC12; Kud18; TESKHKGZ20; SALD20; Kud20; KP20], the graph partitioning problem [UMNM17], systems of polynomial equations [CGHS19; CLT19; RCBPLBMAL22], the prime factorization problem [DA17; JBMHK18; PWHWFCW19; SHIY22], the shortest vector problem [JCLM21; UINKM22; YSFOSMMRT22], and the closest vector problem [YSFOSMMRT22], etc. Also, in [Luc14] many NP problems including Karp's 21 NP-complete problems [Kar72] are reduced to the Ising problem.

Date: January 18, 2023.

Email: treyquantum@gmail.com

The two steps of using quantum annealing to solve an optimization problem are: (1) reduce the optimization problem to the Ising problem; and (2) use a quantum annealer to solve the Ising problem.

1.1. **Ising Problem.** The Ising problem is about finding the lowest energy state of an Ising spin glass. An *Ising spin glass* is a spin glass modeled by the Ising model. A *spin glass* is a magnetic state that has random spins, in contrast to a ferromagnetic solid whose spins are aligned in the same direction. The *Ising model* [Isi25] is a simplified mathematical model of ferromagnetism. Consider an Ising spin glass as a grid Λ of n sites where each site has spin σ_k whose value is either -1 or $+1$. The Ising model only considers interactions between adjacent cites and interactions between each cite and an external magnetic field associating with it. The interaction between two adjacent sites is represented by a real number $J_{i,j} \in \mathbb{R}$; the interaction between each site and its external magnetic field is represented by a real number $h_i \in \mathbb{R}$. The energy of the system is then given by the Hamiltonian (i.e. energy function)

$$H(\sigma) = \sum_{i<j} J_{i,j}\sigma_i\sigma_j + \sum_i h_i\sigma_i. \tag{1}$$

A *configuration* is an assignment of the spin sequence $\sigma = (\sigma_1, \ldots, \sigma_n) \in \{-1,+1\}^n$. The *Ising problem* asks to find a configuration $\sigma \in \{-1,+1\}^n$ that minimizes the Hamiltonian $H(\sigma)$. In the language of physics it means to find the ground state (i.e. lowest energy state) of the Ising model.

1.2. **QUBO Problem.** Note that the Ising problem deals with -1 and $+1$, which is not very natural. We usually work with the quadratic unconstrained binary optimization problem (QUBO). QUBO is equivalent to Ising except that it deals with 0 and 1 instead of -1 and $+1$. Its cost function is

$$H(x) = \sum_{i<j} w_{i,j}x_ix_j + \sum_i v_ix_i, \tag{2}$$

where $w_{i,j}, v_i \in \mathbb{R}$ and $x_i \in \{0,1\}^n$. The problem asks to find $x \in \{0,1\}^n$ that minimizes $H(x)$. The equivalence between Equations (1) and (2) can be easily seen by writing $\sigma_i = 2x_i - 1$.

The QUBO function $H(x)$ can be represented by an upper triangular matrix Q as

$$H(x) = x^\top Q x,$$

where Q is called the QUBO matrix.

The following example shows how to find an Ising/QUBO formulation for an optimization problem.

Example 1. Let $f(x_0,x_1) = x_0 \oplus x_1$, where $\oplus$ is the XOR operation. The optimization problem asks to find a pair $(x_0,x_1) \in \{0,1\}^2$ such that $f(x_0,x_1) = 1$. Note that $f(x_0,x_1) = 1$ if and only if precisely one of x_0 and x_1 is 1. Hence we can define the cost function to be

$$H(x_0,x_1) = (x_0 + x_1 - 1)^2.$$

Whenever H reaches its minimal value 0, the pair (x_0, x_1) is a solution to the optimization problem. Now since $x_0, x_1 \in \{0, 1\}$, we have $x_0 = x_0 x_0$ and $x_1 = x_1 x_1$. Therefore

$$\begin{aligned} H(x_0, x_1) &= (x_0 + x_1 - 1)^2 \\ &= x_0^2 + x_1^2 + 2x_0x_1 - 2x_0 - 2x_1 + 1 \\ &= x_0x_0 + x_1x_1 + 2x_0x_1 - 2x_0x_0 - 2x_1x_1 + 1 \\ &= -x_0x_0 - x_1x_1 + 2x_0x_1 + 1. \end{aligned}$$

Observe that the constant term is useless since minimizing $H(x_0, x_1)$ is equivalent to minimizing

$$\widetilde{H}(x_0, x_1) = -x_0x_0 - x_1x_1 + 2x_0x_1.$$

Write the coefficients of $\widetilde{H}$ into a matrix

$$Q = \begin{pmatrix} -1 & 2 \\ 0 & -1 \end{pmatrix},$$

where $Q_{i,j}$ is the coefficient of $x_i x_j$, we have the desired QUBO function to be

$$\begin{aligned} \widetilde{H}(x_0, x_1) &= x^\top Q x \\ &= \sum_{i=0}^{1} \sum_{j=0}^{1} Q_{i,j} x_i x_j. \end{aligned}$$

1.3. **Quantum Annealing.** The basic idea behind simulated/quantum annealing is the fundamental rule of nature that energy tends to transfer from a less random system to a more random system. This means that things in a higher energy state have the tendency to go to the lowest energy state. So if we could translate the cost function of an optimization problem into the energy function of a physical system, then nature will automatically seek an optimal solution to the optimization problem as it seeks the lowest energy state of the physical system.

Quantum annealing is similar to simulated annealing but with thermal activation replaced by quantum tunneling [KN98]. To go from a high energy state to the lowest energy state, simulated annealing uses temperature to drive the process, while quantum annealing uses a magnetic field; simulated annealing climbs up and down hills to find the lowest "valley", while quantum annealing "digs" a tunnel through the hill and go directly from one valley to another (so-called quantum tunneling); simulated annealing stops when the system is in the lowest temperature, while quantum annealing stops when the system is in the lowest magnetic field.

Specifically, let H_0 be a Hamiltonian that is easy to prepare; and let H_P be the problem Hamiltonian. We prepare a system that is in the ground state of H_0 and let the time dependent Hamiltonian

$$H(t) = \left(1 - \frac{t}{T}\right) H_0 + \frac{t}{T} H_P$$

adiabatically (i.e. very slowly) evolve according to the Schrödinger equation

$$i \frac{d}{dt} |\psi(t)\rangle = H(t) |\psi(t)\rangle$$

until reaching the ground state of the problem Hamiltonian $H_P = H(t = T)$. In the end, the adiabatic theorem [BF28] guarantees that the ground state after time $t = T$ carries an optimal solution to P and we can extract it by measuring the quantum state.

Here H_0 is chosen to be consist of transverse magnetic fields [BASCL13]:

$$H_0 = -h_0 \sum_{i=1}^{n} \sigma_i^x,$$

where $\sigma_i^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ is a Pauli matrix; and H_P is the quantum version of the Hamiltonian (1):

$$\begin{aligned} H_P &= H(\sigma_1^z, \dots, \sigma_n^z) \\ &= \sum_{i<j} J_{i,j} \sigma_i^z \sigma_j^z + \sum_i h_i \sigma_i^z, \end{aligned}$$

where $\sigma_i^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ is a Pauli matrix acting on the i-th qubit of n qubits $\{|+\rangle, |-\rangle\}^{\otimes n}$.

2. Quantum Annealing for Subset Product

The subset product problem (SP) [GJ79, p. 224] asks to solve the exponential equation

$$\prod_{i=1}^{n} a_i^{x_i} = a$$

for a binary vector $x = (x_1, \dots, x_n) \in \{0,1\}^n$, where $a_1, \dots, a_n, a \in \mathbb{Z}$.

In [Li22b] and [Li22c] Li gave classical algorithms for the problem. However those algorithms only make sense when the number of bases $a_1, \dots, a_n$ is not much greater than the number of prime factors $p_1, \dots, p_m$ of the bases $a_1, \dots, a_n$. I.e. $n < m$ or $n \gtrapprox m$. For the regime $n \gg m$, the algorithms can be less efficient than exhaustive search.[1]

Now we give a quantum algorithm for all parameter regimes $n \lesseqgtr m$. We reduce subset product to the Ising problem and apply quantum annealing to search for a solution. The main idea is the following reduction chain:

$$\text{SP} \le_{\text{E}} \text{M-MSS} \le \text{LLS} \le \text{B-LLS} \le \text{Ising/QUBO},$$

where M-MSS is the multiple modular subset sum problem, LLS is the linear least solutions problem, B-LLS is the binary linear least solutions problem, $\le$ denotes "Karp reduces to", and $\le_{\text{E}}$ means that the reduction is empirical.

2.1. SP $\le_{\mathbf{E}}$ M-MSS.

The multiple modular subset sum problem (M-MSS) asks to solve a system of modular linear equations

$$\left\{ \sum_{j=1}^{n} \alpha_{i,j} x_j = \beta_i \pmod{\ell_i} \right\}_{i \in [k]}$$

for a binary solution $x = (x_1, \dots, x_n) \in \{0,1\}^n$.

We show how to reduce an SP instance $(a_1, \dots, a_n, a) \in \mathbb{Z}^{n+1}$ to an M-MSS instance.

[1] This is because at the end of the algorithms we need to search for a binary solution by brute forcing the solution set of a matrix equation $Ax = b \pmod{\ell}$, where ℓ is a prime. Note that there are $\ge \ell^{n-m}$ integral solutions in the solution set. Hence if $\ell > 2$ and $n \gg m$, the size ℓ^{n-m} of the solution set can be greater than the number 2^n of all possible binary strings $x \in \{0,1\}^n$.

Choose $k \geq n$ distinct random small primes $\ell_1, \ldots, \ell_k$ that are polynomial in n. Find k prime moduli $q_i = s_i \ell_i + 1$ that do not divide $a_1, \ldots, a_n$.

Find a generator g_i of the order ℓ_i subgroup $G_i = \left\{ g_i^0, \ldots, g_i^{\ell_i - 1} \right\}$ of $\mathbb{Z}_{q_i}^\times$, for $i = 1, \ldots, k$. This can be done efficiently since any element $a^{(q_i-1)/\ell_i}$ not equals 1 is coprime to q_i and thus is a generator of G_i, for $a \in \mathbb{Z}_{q_i}^\times$.

Find $\alpha_{i,j} \in \mathbb{Z}_{\ell_i}$ such that

$$g_i^{\alpha_{i,j}} = a_j^{(q_i-1)/\ell_i} \pmod{q_i},$$

for $i = 1, \ldots, k$, $j = 1, \ldots, n$. Find β_i such that

$$g_i^{\beta_i} = a^{(q_i-1)/\ell_i} \pmod{q_i},$$

for $i = 1, \ldots, k$. These can be done efficiently since $\alpha_{i,j}$ and β_i are elements of the *small* subgroup $\mathbb{Z}_{\ell_i}$ of $\mathbb{Z}_{q_i-1}$; we can find them by simply brute forcing the small subgroups.

We then have a system of modular equations

$$\left\{ \sum_{j=1}^{n} \alpha_{i,j} x_j = \beta_i \pmod{\ell_i} \right\}_{i \in [k]}.$$

We claim that there exists a polynomial size number $k \in \mathbb{N}$ such that all solutions to the M-MSS are solutions to the SP. This is based on the natural intuition that n random linear equations has high probability to be independent and can constraint the solution set of the M-MSS to be exactly the solution set of the SP.

Experiments. We give experimental results to support our claim. The program is written in Sage and run on a laptop. The laptop details are: MacBook Air 13-inch, early 2015; Processor 1.6 GHz Intel core i5; Memory 8 GB 1600 MHz DDR3; System OS XEI Capitan version 10.11.5.

A difference between the experiments and the real reduction is the choice of the subgroup orders $\ell_1, \ldots, \ell_k$. In the experiments ℓ_i are not "random" primes but the next k primes after the greatest prime factor p_m of the bases $a_1, \ldots, a_n$ of the SP instance $(a_1, \ldots, a_n, X)$.[2] It turns out that this setting is enough for the claim to hold.

The experimental results are shown in the following table, where the column "n" means that each SP has n bases $a_1, \ldots, a_n$; the column "m" means that all $a_1, \ldots, a_n$ (hence each a_i) has at most m prime factors $p_1, \ldots, p_m$; the column "e" means that the exponents of the prime factorizations of $a_1, \ldots, a_n$ are upper bounded by e, i.e. in $\{0, \ldots, e\}$; the column "k" means that we use k MSS instances to represent the SP instance; the column "#Trials" is the numbers of random SP instances we tested for each combination (n, m, d, k); the column "#Successes" is the numbers of trials where the M-MSS instance successfully represents the SP instance, i.e., the solution set of the M-MSS instance is exactly the solution set of the SP instance.

[2] This is to ensure that the moduli $q_i := s_i \ell_i + 1$ are not factors of $a_1, \ldots, a_n$ so that the useless relation $\prod_{i=1}^n a_i^{x_i} = 0 \pmod{q_i}$ will not be created. Also, in the experiments the prime factors of the bases are known to us, but the experiments are just for testing the correctness of the reduction, it does not mean that the reduction requires to find these prime factors.

	n	m	e	k	#Successes	#Trials
1	6	3	1	1,2,3,4,5,6	3,17,12,100,100,100	100
2	7	3	1	1,2,3,4,5,6,7	0,5,8,100,100,100,100	100
3	8	3	1	1,2,3,4,5,6,7,8	0,4,5,100,100,100,100,100	100
4	9	3	1	1,2,3,4,5,6,7,8,9	0,1,3,100,100,100,100,100,100	100
5	6	4	1	1,2,3,4,5,6	5,10,100,100,100,100	100
6	7	4	1	1,2,3,4,5,6,7	0,3,100,100,100,100,100	100
7	8	4	1	1,2,3,4,5,6,7,8	0,4,100,100,100,100,100,100	100
8	9	4	1	1,2,3,4,5,6,7,8,9	0,3,100,100,100,100,100,100,100	100
9	6	2,3,4,5,6,12	1,2,3,6,12	6	100	100
10	7	2,3,4,5,6,7,12	1,2,3,6,12	7	100	100
11	8	2,3,4,5,6,7,8,12	1,2,3,6,12	8	100	100
12	9	2,3,4,5,6,7,8,9,12	1,2,3,6,12	9	100	100

TABLE 1. Test whether M-MSS represents SP.

From Line 1-8 we see that when the number k of MSS instances exceeds some number that is smaller than n, M-MSS starts to represent SP well. This is reasonable because an SP can have multiple solutions and that there probably does not exist n many "independent" linear equations (over the ring $\mathbb{Z}_{\prod_{i=1}^{k}\ell_i}$)[3] corresponding to a solution set of size > 1. Hence we mostly do not need n equations to fix the solution set.

From 9-12 we see that when the number k of MSS instances arrives n, M-MSS always represents SP perfectly, for all tested parameters. This is a consistent result because even in the worst case where the SP has only one solution, it is reasonable to expect that n "random" linear equations is enough to provide sufficiently many "independent" relations to fix the solution.

Note that solving for an integral solution to a modular linear system is easy using the multivariable Chinese remainder theorem [Kni12]. But finding a binary solution is nontrivial. We handle this issue by reducing the M-MSS to the binary linear least squares problem.

2.2. M-MSS ≤ LLS.

We first reduce the M-MSS to the linear least squares problem. The linear least squares problem (LLS) is given a matrix $A \in \mathbb{R}^{m\times n}$ and a vector $b \in \mathbb{R}^m$ and asks to find a vector $z \in \mathbb{R}^n$ that minimizes the Euclidean norm $||Az - b||$.

From the M-MSS we create an LLS as

$$\left(A = \begin{pmatrix} \alpha_{1,1} & \dots & \alpha_{1,n} & \ell_1 & & \\ \vdots & & \vdots & & \ddots & \\ \alpha_{k,1} & \dots & \alpha_{k,n} & & & \ell_k \end{pmatrix}, b = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_k \end{pmatrix}\right), \tag{3}$$

which asks to find a vector $z = (x, y) \in \mathbb{Z}^n \times \mathbb{Z}^k$ that minimizes $||Az - b||$ to 0.

[3] Note that a system of modular equations with different moduli $\ell_1, \dots, \ell_k$ can be converted into a system of modular equations with the same modulus $\prod_{i=1}^{k}\ell_i$. For example, the system $\{2x + 3y = 3 \pmod 5;\ 3x + 4y = 4 \pmod 7\}$ with different moduli 5 and 7 can be converted into the system $\{14x + 21 = 21 \pmod{35};\ 15x + 20y = 20 \pmod{35}\}$ with the same modulus 35 in the following way: $\{(2\cdot7)x + (3\cdot7)y = (3\cdot7) \pmod{5\cdot7};\ (3\cdot5)x + (4\cdot5)y = (4\cdot5) \pmod{4\cdot5}\}$. Also notice that $\mathbb{Z}_{\prod_{i=1}^{k}\ell_i}$ is a ring and not a field. Hence the word "independent" is in the sense of over a ring and not a field.

2.3. **LLS ≤ B-LLS.** The binary linear least squares problem (B-LLS) is given a matrix $A \in \mathbb{R}^{m\times n}$ and a vector $b \in \mathbb{R}^m$ and asks to find a binary vector $z \in \{0,1\}^n$ that minimizes $||Az - b||$.

We reduce the LLS instance to a B-LLS instance since there is a simple transformation from B-LLS into QUBO [OV16] and B-LLS helps to restrict the solution x to be binary.

A generic method to convert LLS to B-LLS is in [OV16], also explained in [BL19]; and a similar trick is used in [YSFOSMMRT22]. We show how to apply it "compactly" (i.e. saving bits as many as possible) to our specific problem.

We will convert the LLS $(A,b) \in \mathbb{Z}^{k\times(n+k)} \times \mathbb{Z}^k$ with solutions $z = (x,y) \in \mathbb{Z}^n \times \mathbb{Z}^k$ to a B-LLS $(\bar{A},t) \in \mathbb{Z}^{k\times(n+d)} \times \mathbb{Z}^k$ with solutions $\bar{z} = (\bar{x},\bar{y}) \in \{0,1\}^n \times \{0,1\}^d$, for some $d > k$.

Let $d_i = \left\lceil \log_2 \frac{n(\ell_i - 1)}{\ell_i} \right\rceil$ and $d = \sum_{i=1}^k d_i$.[4] We let $\bar{x} \in \{0,1\}^n$ and extend $y = (y_1,\ldots,y_k) \in \mathbb{Z}^k$ to

$$\bar{y} = (y_{1,1},\ldots,y_{1,d_1},y_{2,1},\ldots,y_{2,d_2},\ldots,y_{k,1},\ldots,y_{k,d_k}) \in \{0,1\}^d$$

by replacing each $y_i = \sum_{s=1}^{d_i} y_{i,s} 2^{d_i - s}$ by its binary representation $(y_{i,1},\ldots,y_{i,d_i}) \in \{0,1\}^{d_i}$.

We then extend $A \in \mathbb{Z}^{k\times(n+k)}$ into $\bar{A} \in \mathbb{Z}^{k\times(n+d)}$ as follows. For $i = 1,\ldots,k$, we duplicate the $(n+i)$-th column of A to d_i columns and then multiply these d_i columns by $2^{d_i-1},\cdots,2^0$ respectively.

Then $(\bar{A},b)$ is a B-LLS instance with solution(s) $\bar{z} = (\bar{x},\bar{y}) \in \{0,1\}^n \times \{0,1\}^d$.

We call $\bar{A}$ the *extending matrix* of A, $\bar{z}$ the *binary vector* of z, and d_i the *extending ratio* of the column A_{n+i} of A being extended.

2.4. **B-LLS ≤ Ising/QUBO.** Now we can use the simple conversion in [OV16, Section IV] to reduce the B-LLS instance $(\bar{A},b)$ to the QUBO function as

$$H(\bar{z}) = \sum_{r<s} w_{r,s}\bar{z}_r\bar{z}_s + \sum_r v_r \bar{z}_r, \tag{4}$$

where

$$\begin{aligned} w_{r,s} &= 2\sum_i \bar{A}_{i,r}\bar{A}_{i,s}, \\ v_r &= \sum_i \bar{A}_{i,r}(\bar{A}_{i,r} - 2b_r), \end{aligned}$$

and $r,s \in [n+d]$.

Having this formulation, one can then run a quantum annealer (such as the D-Wave series) to find an optimal solution $\bar{z} = (\bar{x},\bar{y})$ to the QUBO, where $\bar{x}$ is the solution to our SP.

2.5. **Subset Product Algorithm.** In sum of the above reductions, the algorithm for subset product is the following.

[4] Here we set $d_i = \left\lceil \log_2 \frac{n(\ell_i-1)}{\ell_i} \right\rceil$ because y_i satisfies $\sum_{j=1}^n \alpha_{i,j}x_j = \beta_i + y_i\ell_i$ and we must have $y_i \le \frac{n(\ell_i-1)}{\ell_i}$.

Algorithm 1 Quantum Annealing for Subset Product

```
Input: A subset product instance $(a_1,\dots,a_n,a)\in\mathbb{Z}^{n+1}$.
Output: A vector $x=(x_1,\dots,x_n)\in\{0,1\}^n$ such that $\prod_{j=1}^n a_j^{x_j}=a$.
1: Sample $k\ge n$ distinct random primes $\ell_1,\dots,\ell_k$ of size polynomial in $n$ and choose $k$ prime moduli $q_1,\dots,q_k$ of the form $q_i=s_i\ell_i+1$ such that $q_i\nmid a_j$ for all $i\in[k]$, $j\in[n]$;
2: Find a generator $g_i$ of the order $\ell_i$ subgroup $G_i$ of $\mathbb{Z}_{q_i}^\times$ and find $\alpha_{i,j}$ such that $g_i^{\alpha_{i,j}}=a_j^{s_i}$ (mod $q_i$) as well as $\beta_i$ such that $g_i^{\beta_i}=a^{s_i}$ (mod $q_i$), by enumerating elements in $\mathbb{Z}_{\ell_i}$, for all $i\in[k]$, $j\in[n]$;
3: Define an LLS instance $(A,b)$ as Equation (3);
4: Compute the extending matrix $\bar{A}$ of $A$;
5: Define a QUBO function $H(\bar{z})$ as Equation (4);
6: Use a quantum annealer to solve the QUBO $H(\bar{z})$ for a vector $\bar{z}=(\bar{x},\bar{y})\in\{0,1\}^n\times\{0,1\}^{\sum_{i=1}^k\left\lceil\log_2\frac{n(\ell_i-1)}{\ell_i}\right\rceil}$;
7: Return $\bar{x}$.
```

The complexity of Step 1-5 is polynomial in n. The efficiency of the algorithm mainly depends on the efficiency of the quantum annealer.

3. Quantum Annealing for Noisy Subset Product

Note that there are many abstract versions of the noisy subset product problem [Li22d; Li22h]. The concrete versions used in building key exchange scheme [Li22d], public-key cryptosystem [Li22e] and digital signature [Li22f] are over integers and with concrete noise distributions. We consider a general version of the concrete versions, i.e., the noisy subset product problem over integers but with arbitrary distributions of the bases and the errors, and possibly with different moduli.

Let $D_{1,i}(\mathbb{Z}_{q_i})$ be a distribution over $\mathbb{Z}_{q_i}$; and let $D_2(L)$ be a distribution over a set L of positive integers. The noisy subset product problem (NSP) asks to solve the exponential equation system

$$\left\{\left(\prod_{j=1}^{n}a_{i,j}^{x_j}\right)\cdot e_i=a_i\pmod{q_i}\right\}_{i\in[k]}$$

for a pair $(x,e)=(x_1,\dots,x_n,e_1,\dots,,e_k)\in\{0,1\}^n\times L^k$, where q_i are safe primes, $a_{i,j}\leftarrow D_{1,i}(\mathbb{Z}_{q_i})$ and $a_i=\prod_{j=1}^n a_{i,j}^{x_j}\cdot e_i \mod q_i$ are given by the problem, but $e_i\leftarrow D_2(L)$ are not given. If k is large enough, the solution (x,e) is unique with overwhelming probability [Li22d].

We prove the following reduction chain:

$$\text{NSP}\le\text{NSS}\le\text{LLS}\le\text{B-LLS}\le\text{Ising/QUBO},$$

where NSS is the noisy subset sum problem.

3.1. **NSP ≤ NSS.** Let $D_{1,i}(\mathbb{Z}_{N_i})$ be a distribution over $\mathbb{Z}_{N_i}$; and let $D_2(S)$ be a distribution over a set S of positive integers. The noisy subset sum problem (NSS) [Li22d; Li22h] asks to solve the linear system

$$\left\{\sum_{j=1}^{n} \alpha_{i,j} x_j + \epsilon_i = \beta_i \pmod{N_i}\right\}_{i\in[k]}$$

for a pair $(x,\epsilon) = (x_1,\dots,x_n,\epsilon_1,\dots,\epsilon_k) \in \{0,1\}^n \times S^k$, where $N_i \geq 2$ are positive integers, $\alpha_{i,j} \leftarrow D_{1,i}(\mathbb{Z}_{N_i})$ and $\beta_i = \sum_{j=1}^{n} \alpha_{i,j} x_j \cdot \epsilon_i \mod N_i$ are given by the problem, but $\epsilon_i \leftarrow D_2(S)$ are not given. If k is large enough, the solution (x,ϵ) is unique with overwhelming probability.

We reduce NSP to NSS with $N_1 = \cdots = N_k = 2$. Simply take the Legendre symbols for the exponential equations

$$\left(\prod_{j=1}^{n} a_{i,j}^{x_j}\right) \cdot e_i = a_i \pmod{q_i}$$

to get the equations

$$\prod_{j=1}^{n} \left(\frac{a_{i,j}}{q_i}\right)^{x_j} \cdot \left(\frac{e_i}{q_i}\right) = \left(\frac{a_i}{q_i}\right);$$

and then extract the parity-check linear equations

$$\sum_{j=1}^{n} \alpha_{i,j} x_j + \epsilon_i = \beta_i \pmod{2},$$

where $\alpha_{i,j} = (1-(a_{i,j}/q_i))/2$, $\beta_i = (1-(a_i/q_i))/2$, $i = 1,\dots,k$. Write in the matrix form we have

$$Ax + \epsilon = b \pmod{2},$$

where $A = \{\alpha_{i,j}\} \in \mathbb{Z}_2^{k\times n}$, $\epsilon \in \mathbb{Z}_2^k$ and $b = (\beta_i) \in \mathbb{Z}_2^k$.

It is obvious that the NSS instance and the NSP instance share the same solution(s).

3.2. **NSS ≤ LLS.** Now we define an LLS as

$$(B = (A, I_k, 2I_k), b)$$

which asks to find a vector $z = (x,\epsilon,c) \in \mathbb{Z}_2^n \times \mathbb{Z}_2^k \times \mathbb{Z}_{\lceil(n+1)/2\rceil}^k$ that minimizes $||Bz - b||$ to 0. Note that $||Bz - b|| = 0$ only when $Ax + I_k\epsilon + 2I_k c = b$ hence $Ax + \epsilon = b \pmod{2}$. Also when k is large enough, the NSS has a unique solution with overwhelming probability. Hence the pair (x,ϵ) extracted from the LLS solution v is THE NSS solution with overwhelming probability.

3.3. **LLS ≤ B-LLS.** Let $\bar{B} = (A, I_k, \overline{2I_k})$ be the extending matrix of B. Then $(\bar{B}, b)$ is a B-LLS that asks for a vector $\bar{z} = (x,\epsilon,\bar{c}) \in \mathbb{Z}_2^n \times \mathbb{Z}_2^k \times \mathbb{Z}_2^{k\lceil\log_2((n+1)/2)\rceil}$ that minimizes $||\bar{B}\bar{z} - b||$ to 0. Here the parameter $\lceil\log_2((n+1)/2)\rceil$ is because each element c_i of c satisfies $-(n+1)/2 \leq c_i \leq (n+1)/2$ and that $\lceil\log_2((n+1)/2)\rceil$ is the tight bit length for it.

3.4. **B-LLS ≤ Ising/QUBO.** The QUBO function of the B-LLS instance $(\bar{B}, b)$ is

$$H(\bar{z}) = \sum_{r<s} w_{r,s} \bar{z}_r \bar{z}_s + \sum_r v_r \bar{z}_r, \tag{5}$$

where

$$w_{r,s} = 2\sum_i \bar{B}_{i,r}\bar{B}_{i,s},$$
$$v_r = \sum_i \bar{B}_{i,r}(\bar{B}_{i,r} - 2b_r),$$

and $r,s \in [n + k + k\lceil\log_2((n+1)/2)\rceil]$.

3.5. **Noisy Subset Product Algorithm.**

In sum of the above reductions, the algorithm for noisy subset product is the following.

Algorithm 2 Quantum Annealing for Noisy Subset Product

```
Input: A noisy subset product instance $\{a_{i,1},\dots,a_{i,n},b_i,q_i,L,D_{1,i},D_2\}_{i\in[k]}$.
Output: A pair $(x,e)\in\{0,1\}^n\times L^k$ such that $\left(\prod_{j=1}^n a_{i,j}^{x_j}\right)\cdot e_i = b_i \pmod{q_i}$ for all $i\in[k]$.
1: Compute $\alpha_{i,j} = \left(1-a_{i,j}^{(q_i-1)/2} \mod q_i\right)/2$ for all $i\in[k]$, $j\in[n]$, denote $A := \{\alpha_{i,j}\}_{i\in[k],j\in[n]}$;
2: Compute $\beta_i = \left(b_i^{(q_i-1)/2} \mod q_i\right)/2$ for all $i\in[k]$, denote $b=(\beta_1,\dots,\beta_k)$;
3: Create a B-LLS instance $(\bar{B}=(A,I_k,\overline{2I}_k),b)$;
4: Cover the B-LLS $(\bar{B},b)$ into a QUBO function $H(\bar{z})$ as Equation (5);
5: Use a quantum annealer to solve the QUBO $H(\bar{z})$ for a vector $\bar{z}=(x,\epsilon,\bar{c})\in\mathbb{Z}_2^n\times\mathbb{Z}_2^k\times\mathbb{Z}_2^{k\lceil\log_2((n+1)/2)\rceil}$;
6: Compute $e_i = g_i^{\epsilon_i} \mod q_i$ for all $i\in[k]$, denote $e := (e_1,\dots,e_k)$;
7: Return $(x,e)$.
```

The complexities of Steps 1, 2, 3, 4 and 6 are polynomial in n. The efficiency of the algorithm mainly depends on the efficiency of the quantum annealer.

4. Lattice Algorithms

Note that in the quantum annealing algorithms we use B-LLS/Ising/QUBO (basically using physics) to restrict the solution to be binary. Now we give another way to achieve this, which is via the closest vector problem in a special lattice. This leads to another two algorithms for SP and NSP via solving the closest vector problem. These algorithms are generally inefficient since the closest vector problem is believed to be hard in the worst case. The meaning of these algorithms is that they are efficient when the closest vector problem falls into the easy regimes of bounded distance decoding problem that can be solved efficiently by the LLL algorithm [LLL82] or Babai's nearest plane algorithm [Bab86]. See [ABCG22] for an overview of algorithms for the bounded distance decoding problem.

Let $\mathbb{R}^m$ be the m-dimensional Euclidean space. A lattice in $\mathbb{R}^m$ is the set $\Lambda(B)=\{Bz : z\in\mathbb{Z}^n\}$, where $B=(b_1,\dots,b_n)\in\mathbb{R}^{m\times n}$ is a matrix with linear independent columns $b_1,\dots,b_n$. We call $\{b_1,\dots,b_n\}$ the *lattice basis*. The positive integers m and n are called the *dimension* and *rank* of the lattice. If $m=n$ then the lattice is called a *full rank* lattice.

The closest vector problem (CVP) is given a (typically full rank) lattice basis $B\in\mathbb{Z}^{m\times n}$ and a target vector $t\in\mathbb{R}^m$, find a lattice vector $Bz\in\Lambda(B)$ closest to t.

4.1. **Lattice Algorithm for Subset Product.**

We show the following reduction chain:

$$\mathrm{SP} \le_{\mathrm{E}} \text{M-MSS} \le \mathrm{CVP}.$$

The reduction from SP to M-MSS is done in Section 2.1. To reduce our M-MSS instance to a CVP instance, simply create a lattice basis:

$$B = (b_1, \ldots, b_{n+k}) = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ \alpha_{1,1} & \ldots & \alpha_{1,n} & \ell_1 & & \\ \vdots & & \vdots & & \ddots & \\ \alpha_{k,1} & \ldots & \alpha_{k,n} & & & \ell_k \end{bmatrix} \in \mathbb{Z}^{(n+k)\times(n+k)}, \tag{6}$$

and a target vector:

$$t = \left(\frac{1}{2}, \ldots, \frac{1}{2}, \beta_1, \ldots, \beta_k\right)^{\top} \in \mathbb{R}^{n+k}. \tag{7}$$

Then (B, t) is a CVP instance.

We claim that any solution to this CVP instance implies a solution to the M-MSS instance. Let Bz be any vector in $\Lambda(B)$ with $z = (x, y) = (x_1, \ldots, x_n, y_1, \ldots, y_k) \in \mathbb{Z}^{n+k}$. We have that

$$||Bz - t||^2 = \sum_{j=1}^{n} \left| x_j - \frac{1}{2} \right|^2 + \sum_{i=1}^{k} \left| \sum_{j=1}^{n} \alpha_{i,j} x_j + \ell_i y_i - \beta_i \right|^2.$$

It is not hard to see that $||Bz - t||^2$ achieves its minimal value $n/4$ if and only if

$$(x_1, \ldots, x_n) \in \{0, 1\}^n$$

and at the same time

$$\sum_{j=1}^{n} \alpha_{i,j} x_j + \ell_i y_i - \beta_i = 0$$

for all $i \in [k]$. In other words, if Bz is a closest vector to t, then x must be a (binary) solution to the M-MSS.

The algorithm is as the following.

Algorithm 3 Lattice Algorithm for Subset Product

```
Input: A subset product instance $(a_1, \ldots, a_n, a) \in \mathbb{Z}^{n+1}$.
Output: A vector $x = (x_1, \ldots, x_n) \in \{0,1\}^n$ such that $\prod_{j=1}^{n} a_j^{x_j} = a$.
1: Sample $k \geq n$ distinct random primes $\ell_1, \ldots, \ell_k$ of size polynomial in $n$ and choose $k$ prime moduli $q_1, \ldots, q_k$ of the form $q_i = s_i \ell_i + 1$ such that $q_i \nmid a_j$ for all $i \in [k]$, $j \in [n]$;
2: Find a generator $g_i$ of the order $\ell_i$ subgroup $G_i$ of $\mathbb{Z}_{q_i}^{\times}$ and find $\alpha_{i,j}$ such that $g_i^{\alpha_{i,j}} = a_j^{s_i} \pmod{q_i}$ as well as $\beta_i$ such that $g_i^{\beta_i} = a^{s_i} \pmod{q_i}$, by enumerating elements in $\mathbb{Z}_{\ell_i}$, for all $i \in [k]$, $j \in [n]$;
3: Define a CVP basis $B$ as Equation (6) and a CVP target vector $t$ as Equation (7);
4: Solve the CVP $(B, t)$ for a vector $z = (x, y) \in \mathbb{Z}^{n+k}$;
5: Return $x$.
```

The complexity of Step 1-3 is polynomial in n. The efficiency of the algorithm mainly depends on the efficiency of the CVP solver on the special CVP instance. In the worst case the CVP is believed to be hard.

4.2. **Lattice Algorithm for Noisy Subset Product.** We show the following reduction chain:

$$\mathrm{NSP} \le \mathrm{NSS} \le \mathrm{CVP}.$$

The reduction from NSP to NSS is done in Section 3.1. We reduce our NSS to the following CVP:

$$\left(B = \begin{pmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ \alpha_{1,1} & \dots & \alpha_{1,n} & 1 & & & q-1 & & \\ \vdots & & \vdots & & \ddots & & & \ddots & \\ \alpha_{k,1} & \dots & \alpha_{k,n} & & & 1 & & & q-1 \end{pmatrix}, t = \begin{pmatrix} 1/2 \\ \vdots \\ 1/2 \\ \beta_1 \\ \vdots \\ \beta_k \end{pmatrix} \right), \tag{8}$$

which asks to find a vector $z = (x, e, c) \in \mathbb{Z}^n \times \mathbb{Z}^k \times \mathbb{Z}^k$ such that $||Bz - t|| = \sqrt{n}/2$. Note that the solution (x, e) to the NSP and NSS is unique with overwhelming probability. Hence the pair (x, e) extracted from the CVP solution z is THE solution to the NSP with overwhelming probability. The algorithm is the following.

Algorithm 4 Lattice Algorithm for Noisy Subset Product

```
Input: A noisy subset product instance $\{a_{i,1},\dots,a_{i,n},b_i,q_i,L,D_{1,i},D_2\}_{i\in[k]}$.
Output: A pair $(x,e) \in \{0,1\}^n \times L^k$ such that $\left(\prod_{j=1}^n a_{i,j}^{x_j}\right) \cdot e_i = b_i \pmod{q_i}$ for all $i \in [k]$.
1: Compute $\alpha_{i,j} = \left(1 - a_{i,j}^{(q_i-1)/2} \mod q_i\right)/2$ for all $i \in [k]$, $j \in [n]$;
2: Compute $\beta_i = \left(b_i^{(q_i-1)/2} \mod q_i\right)/2$ for all $i \in [k]$;
3: Define a CVP $(B,t)$ as Equation (8);
4: Solve the CVP $(B,t)$ for a vector $z \in (x,e,c) \in \mathbb{Z}^n \times \mathbb{Z}^k \times \mathbb{Z}^k$;
5: Return $(x,e)$.
```

The complexity of Step 1-3 is polynomial in n. The efficiency of the algorithm mainly depends on the efficiency of the CVP solver on the special CVP instance. In the worst case the CVP is believed to be hard.

5. Final remarks

If we are allowed to use a factorization algorithm, then subset product can be firstly reduced to exact cover then reduced to Ising using the simple reduction from exact cover to Ising in [Luc14]. However this is a quantum reduction since currently we only have quantum algorithms for factorization [Sho94; Sho99].

If we are allowed to use a discrete logarithm algorithm, then noisy subset product can be firstly reduced to noisy subset sum by taking discrete logarithms then reduced to Ising in a similar way as ours. However this is a quantum reduction since at the moment we only have quantum algorithms for discrete logarithms [Sho94; Sho99].

The efficiencies of our algorithms rely on the efficiencies of the quantum annealing processes or the CVP algorithms. In fact, one of our motivations for finding algorithms for noisy subset product is to understand its hardness as an underlying assumption in cryptography.

From Algorithm 2 and 4, which are currently the only existing algorithms for the problem, it is unlikely that the problem can be solved in polynomial time. This is because solving it is quite similar to solving the learning parity with noise problem (which is over $\mathbb{Z}_2$) or more general noisy subset sum problems (which are over $\mathbb{Z}_N$ with $N > 2$), where the problem over $\mathbb{Z}_2$ is widely believed to be hard [Pie12], and the problems over $\mathbb{Z}_N$ ($N > 2$) are plausibly hard.

References

[ABCG22] Richard Allen, Ratip Emin Berker, Sílvia Casacuberta, and Michael Gul. "Quantum and Classical Algorithms for Bounded Distance Decoding". In: *arXiv preprint arXiv:2203.05019* (2022).

[Bab86] László Babai. "On Lovászlattice reduction and the nearest lattice point problem". In: *Combinatorica* 6.1 (1986), pp. 1–13.

[BASCL13] Sergio Boixo, Tameem Albash, Federico M Spedalieri, Nicholas Chancellor, and Daniel A Lidar. "Experimental signature of programmable quantum annealing". In: *Nature communications* 4.1 (2013), pp. 1–8.

[BF28] Max Born and Vladimir Fock. "Beweis des adiabatensatzes". In: *Zeitschrift für Physik* 51.3 (1928), pp. 165–180.

[BL19] Ajinkya Borle and Samuel J Lomonaco. "Analyzing the quantum annealing approach for solving linear least squares problems". In: *International Workshop on Algorithms and Computation*. Springer. 2019, pp. 289–301.

[CGHS19] Chia Cheng Chang, Arjun Gambhir, Travis S Humble, and Shigetoshi Sota. "Quantum annealing for systems of polynomial equations". In: *Scientific reports* 9.1 (2019), pp. 1–9.

[CLT19] Tyler H Chang, Thomas CH Lux, and Sai Sindhura Tipirneni. "Least-squares solutions to polynomial systems of equations with quantum annealing". In: *Quantum Information Processing* 18.12 (2019), pp. 1–17.

[DA17] Raouf Dridi and Hedayat Alghassi. "Prime factorization using quantum annealing and computational algebraic geometry". In: *Scientific reports* 7.1 (2017), pp. 1–10.

[GJ79] Michael R Garey and David S Johnson. *Computers and intractability*. Vol. 174. freeman San Francisco, 1979.

[Isi25] Ernst Ising. "Contribution to the theory of ferromagnetism". In: *Z. Phys* 31.1 (1925), pp. 253–258.

[JBMHK18] Shuxian Jiang, Keith A Britt, Alexander J McCaskey, Travis S Humble, and Sabre Kais. "Quantum annealing for prime factorization". In: *Scientific reports* 8.1 (2018), pp. 1–9.

[JCLM21] David Joseph, Adam Callison, Cong Ling, and Florian Mintert. "Two quantum Ising algorithms for the shortest-vector problem". In: *Physical Review A* 103.3 (2021), p. 032433.

[Kar72] Richard M Karp. "Reducibility among combinatorial problems". In: *Complexity of computer computations*. Springer, 1972, pp. 85–103.

[KN98] Tadashi Kadowaki and Hidetoshi Nishimori. "Quantum annealing in the transverse Ising model". In: *Physical Review E* 58.5 (1998), p. 5355.

[Kni12] Oliver Knill. “A multivariable Chinese remainder theorem”. In: *arXiv preprint arXiv:1206.5114* (2012).

[KP20] Julia Kwok and Kristen Pudenz. “Graph coloring with quantum annealing”. In: *arXiv preprint arXiv:2012.04470* (2020).

[Kud18] Kazue Kudo. “Constrained quantum annealing of graph coloring”. In: *Physical Review A* 98.2 (2018), p. 022301.

[Kud20] Kazue Kudo. “Localization in the constrained quantum annealing of graph coloring”. In: *Journal of the Physical Society of Japan* 89.6 (2020), p. 064001.

[Li22a] Trey Li. *Subset Product with Errors over Unique Factorization Domains and Ideal Class Groups of Dedekind Domains*. Cryptology ePrint Archive, Paper 2022/1305. https://eprint.iacr.org/2022/1305. 2022, October 1. URL: https://eprint.iacr.org/2022/1305.

[Li22b] Trey Li. *Jacobi Symbol Parity Checking Algorithm for Subset Product*. Cryptology ePrint Archive, Paper 2022/1308. https://eprint.iacr.org/2022/1308. 2022, October 2. URL: https://eprint.iacr.org/2022/1308.

[Li22c] Trey Li. *Power Residue Symbol Order Detecting Algorithm for Subset Product over Algebraic Integers*. Cryptology ePrint Archive, Paper 2022/1310. https://eprint.iacr.org/2022/1310. 2022, October 3. URL: https://eprint.iacr.org/2022/1310.

[Li22d] Trey Li. *Multiple Modular Unique Factorization Domain Subset Product with Errors*. Cryptology ePrint Archive, Paper 2022/1312. https://eprint.iacr.org/2022/1312. 2022, October 4. URL: https://eprint.iacr.org/2022/1312.

[Li22e] Trey Li. *Post-Quantum Key Exchange from Subset Product With Errors*. Cryptology ePrint Archive, Paper 2022/1319. https://eprint.iacr.org/2022/1319. 2022, October 5. URL: https://eprint.iacr.org/2022/1319.

[Li22f] Trey Li. *Post-Quantum Public Key Cryptosystem from Subset Product with Errors*. Cryptology ePrint Archive, Paper 2022/1327. https://eprint.iacr.org/2022/1327. 2022, October 6. URL: https://eprint.iacr.org/2022/1327.

[Li22g] Trey Li. *Post-Quantum Signature from Subset Product with Errors*. Cryptology ePrint Archive, Paper 2022/1334. https://eprint.iacr.org/2022/1334. 2022, October 7. URL: https://eprint.iacr.org/2022/1334.

[Li22h] Trey Li. *Discrete Exponential Equations and Noisy Systems*. Cryptology ePrint Archive, Paper 2022/1344. https://eprint.iacr.org/2022/1344. 2022, October 8. URL: https://eprint.iacr.org/2022/1344.

[Li22i] Trey Li. *Generic Signature from Noisy Systems*. Cryptology ePrint Archive, Paper 2022/1346. https://eprint.iacr.org/2022/1346. 2022, October 9. URL: https://eprint.iacr.org/2022/1346.

[LLL82] Arjen Klaas Lenstra, Hendrik Willem Lenstra, and László Lovász. “Factoring polynomials with rational coefficients”. In: *Mathematische Annalen* 261.4 (1982), pp. 515–534.

[LM09] Vadim Lyubashevsky and Daniele Micciancio. “On Bounded Distance Decoding, Unique Shortest Vectors, and the Minimum Distance Problem”. In: *Advances in Cryptology - CRYPTO 2009*. Ed. by Shai Halevi. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 577–594. ISBN: 978-3-642-03356-8.

[Luc14] Andrew Lucas. “Ising formulations of many NP problems”. In: *Frontiers in physics* (2014), p. 5.

[MST04] Roman Martoňák, Giuseppe E Santoro, and Erio Tosatti. “Quantum annealing of the traveling-salesman problem”. In: *Physical Review E* 70.5 (2004), p. 057701.

[OV16] Daniel O’Malley and Velimir V Vesselinov. “Toq. jl: A high-level programming language for d-wave machines based on julia”. In: *2016 IEEE High Performance Extreme Computing Conference (HPEC)*. IEEE. 2016, pp. 1–7.

[Pie12] Krzysztof Pietrzak. “Cryptography from learning parity with noise”. In: *International Conference on Current Trends in Theory and Practice of Computer Science*. Springer. 2012, pp. 99–114.

[PWHWFCW19] WangChun Peng, BaoNan Wang, Feng Hu, YunJiang Wang, XianJin Fang, XingYuan Chen, and Chao Wang. “Factoring larger integers with fewer qubits via quantum annealing with optimized parameters”. In: *SCIENCE CHINA Physics, Mechanics & Astronomy* 62.6 (2019), pp. 1–8.

[RCBPLBMAL22] Sergi Ramos-Calderer, Carlos Bravo-Prieto, Ruge Lin, Emanuele Bellini, Marc Manzano, Najwa Aaraj, and José I Latorre. “Solving systems of Boolean multivariate equations with quantum annealing”. In: *Physical Review Research* 4.1 (2022), p. 013096.

[RCC89] Pulak Ray, Bikas K Chakrabarti, and Arunava Chakrabarti. “Sherrington-Kirkpatrick model in a transverse field: Absence of replica symmetry breaking due to quantum fluctuations”. In: *Physical Review B* 39.16 (1989), p. 11828.

[SALD20] Carla Silva, Ana Aguiar, Priscila Lima, and Inês Dutra. “Mapping graph coloring to quantum annealing”. In: *Quantum Machine Intelligence* 2.2 (2020), pp. 1–19.

[SHIY22] Daisuke Saida, Mutsuo Hidaka, Kentaro Imafuku, and Yuki Yamanashi. “Factorization by quantum annealing using superconducting flux qubits implementing a multiplier Hamiltonian”. In: *Scientific reports* 12.1 (2022), pp. 1–8.

[Sho94] Peter W Shor. “Algorithms for quantum computation: discrete logarithms and factoring”. In: *Proceedings 35th annual symposium on foundations of computer science*. Ieee. 1994, pp. 124–134.

[Sho99] Peter W Shor. “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer”. In: *SIAM review* 41.2 (1999), pp. 303–332.

[TC11a] Olawale Titiloye and Alan Crispin. “Graph coloring with a distributed hybrid quantum annealing algorithm”. In: *KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications*. Springer. 2011, pp. 553–562.

[TC11b] Olawale Titiloye and Alan Crispin. “Quantum annealing of the graph coloring problem”. In: *Discrete Optimization* 8.2 (2011), pp. 376–384.

[TC12] Olawale Titiloye and Alan Crispin. “Parameter tuning patterns for random graph coloring with quantum annealing”. In: *PloS one* 7.11 (2012), e50060.

[TESKHKGZ20] Zsolt Tabi, Kareem H El-Safty, Zsófia Kallus, Péter Hága, Tamás Kozsik, Adam Glos, and Zoltán Zimborás. “Quantum optimization for the graph coloring problem with space-efficient embedding”. In: *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE. 2020, pp. 56–62.

[UINKM22] Katsuki Ura, Takashi Imoto, Tetsuro Nikuni, Shiro Kawabata, and Yuichiro Matsuzaki. “Analysis of the shortest vector problems with the quantum annealing to search the excited states”. In: *arXiv preprint arXiv:2209.03721* (2022).

[UMNM17] Hayato Ushijima-Mwesigwa, Christian FA Negre, and Susan M Mniszewski. “Graph partitioning using quantum annealing on the d-wave system”. In: *Proceedings of the Second International Workshop on Post Moores Era Supercomputing*. 2017, pp. 22–29.

[YSFOSMMRT22] Junpei Yamaguchi, Toshiya Shimizu, Kazuyoshi Furukawa, Ryuichi Ohori, Takeshi Shimoyama, Avradip Mandal, Hart Montgomery, Arnab Roy, and Ohwa Takuya. “ANNEALING-BASED ALGORITHM FOR SOLVING CVP AND SVP”. In: *Journal of the Operations Research Society of Japan* 65.3 (2022), pp. 121–137.