CIS 410 Final Report on Hidden Subgroup Problem

Zhimeng Wang, Dongmin Roh, Matthew Jagielski

June 8, 2016

1 Motivation

The hidden subgroup problem (HSP) is an computational algebra problem which has been shown to have a lot of interesting consequences and motivations. For example, the *Shor's quantum algorithm* of factoring integers and solving the discrete logarithm problem can be reduced to solving the HSP on finite abelian groups.

Definition 1. Given a group G, a subgroup $H \leq G$, and a set X, a function $f : G \to X$ **hides** H if $\forall g_1, g_2 \in G, f(g_1) = f(g_2)$ iff $g_1H = g_2H$, that is, g_1, g_2 are in the same coset of H.

Definition 2. Now, the **Hidden Subgroup Problem (HSP)** is a problem with inputs: a group G, a set X, and a function oracle $f : G \to X$ hiding a subgroup H. The function oracle uses $\log(\mid G \mid + \mid X \mid)$ bits. The desired output is a generating set of H.

It is known that there exits a quantum algorithm which solves with certainty a hidden subgroup problem of an arbitrary finite group in a polynomial (in $log|G|$) number of calls to the oracle. In addition, quantum computers have been shown to have very good speedups for some instances of the problem. In fact, because quantum computers can factor integers much faster than classical computers can, quantum computers can solve the HSP on finite abelian groups in polynomial time.

Two unknowns regarding the HSP are whether the symmetric group and the dihedral group have efficient quantum algorithms for solving HSP. If an efficient quantum algorithm were to be found for the symmetric group HSP, we would have an efficient algorithm for *graph isomorphism*, a very important problem in theoretical computer science and for Eugene Luks. A polynomial time dihedral group HSP algorithm would give a polynomial time algorithm for solving the *shortest vector problem on lattices*, a problem which is...(line truncated)...

Our group has some background in abstract algebra and algebraic number theory, so this is an attractive topic for us to explore. Also, one of us is studying the shortest vector problem for his undergraduate thesis, so this is of increased interest.

2 Quantum Complexity Results

We realize that many of the well-known quantum algorithms such as Deutsch, Simon, Order-finding, Integer factorization, etc are instances of the hidden subgroup problem. Using an algorithm similiar to these problems, we'll see that for a finite abelian group we can solve the HSP efficiently.

Although no efficient quantum algorithm for non-abelian groups fulfills polynomial time complexity, the quantum query complexity is polynomial. This is exponentially better than any classical algorithms.

2.1 Quantum Time Complexity of an Abelian Group

We give an outline of the algorithm that solves the HSP for an abelian group G in a polynomial number of operations in $log|G|$. This algorithm solves with bounded error; we can find a subset that generates the hidden subgroup with probability at least $\frac{2}{3}$. It is worth mentioning that for Abelian groups of smooth order[1], this success probability can be improved to one.

Recall that finite Abelian groups are isomorphic to a product of finite cyclic groups of prime power order. This allows us to perform the quantum fourier transfrom of f over G in an efficient way. (Decomposing an arbitrary abelian group is a difficult problem. However, Shor's quantum algorithm for factorizing integers can be applied to decompose an abelian group efficiently)

Suppose that an abelian group G is isomorphic to a product of cyclic groups $\mathbb{Z}_{t_1} \times \mathbb{Z}_{t_2} \times \cdots \times \mathbb{Z}_{t_k}$. Consider applying the operation $F_G = F_{t_1} \otimes F_{t_2} \otimes \cdots \otimes F_{t_k}$, which performs a quantum fourier transform to each ith state, to some element $g \in G$ (Here, $|g\rangle = |g_1\rangle |g_2\rangle \cdots |g_k\rangle$). Then, we attach an ancilla bit and send the state through the black box to obtain

$$\frac{1}{\sqrt{|G|}} \sum_{g \in G} |g\rangle \, |f(g)\rangle$$

We perform a measurement on the second register to get some value $f(g^{'})$. This leaves the first register to be in the state

$$\frac{1}{\sqrt{|H|}} \sum_{h \in H} \left| g^{'} + h \right\rangle$$

where H is our hidden subgroup. We apply the operation F_G once again then measure this first register. With certainty we get a uniformly random element of $H^{\perp}$, which is a dual group of H.[2] Thus, repeating these steps about $n = \lceil log|G| \rceil$ many times gives with high probability a set of generators $g^1, \ldots, g^{O(n)}$ of $H^{\perp}$. The linear congruences corresponding to these elements determine H; randomly pick a solution to each linear congruence to get a uniformly random element of H, repeat this $O(n)$ times gives us a set of generators of H with high probabilty.

[1] The order of a group G is smooth if all prime factors are at most $log^c|G|$ for some fixed constant c

[2] The dual is defined as $H^{\perp} = \{g \in G : \chi_g(h) = 1, \forall h \in H\}$ where $\chi_g(h) = \prod_{i=1}^{k} e^{\frac{2\pi i g_i h_i}{t_i}}$

2.2 Quantum Query Complexity of a Finite Group

Our motivation is to find an efficient quantum algorithm which can solve the HSP for any arbitrary finite group G in a polynomial calls to the given oracle. Given r many distinct subgroups of G, we are looking for a generating set for one of the subgroups. We can assume that any algorithms for the HSP always output a subset of a subgroup H; if an algorithm outputs some subset $X \nsubseteq H$, we simply find the intersection of X with H by keeping $x \in X$ only if $f(x) = f(1_G)$.

Let f be a function satisfying the conditions of the HSP. Fix an ordering of the distinct subgroups $H_1, H_2, \ldots, H_r$ such that $|H_i| \geq |H_{i+1}|$ for all $1 \leq i \leq r$.[3] Also let $N = |G|$ and consider $n = log|G|$ to be the input size. [4]

Theorem 3. *There exists a quantum algorithm that solves the HSP for any finite group G in $O(log^4|G|)$ calls to the oracle. The algorithm has exponentially small error probability in $log|G|$.*

The algorithm considers $2 + 2s$ registers, where s is a positive integer chosen to lower the error probability: 1st contains a subgroup index $1 \leq v \leq r$, 2nd contains a counter $1 \leq l \leq r$, remaining $2s$ are pairs of couplets so that in each couplet the first contains an element of G and the second some image of f. We call the first register in a couplet as a "subgroup" register and the second as a "function" register.

We say that a function f is H-*periodic* if f is constant of the left cosets of a subgroup H of G. H being a *hidden subgroup* of f is an instance of f taking distinct values on distinct cosets of H.

A left *translation* of a subgroup H is a subset $T \subseteq G$ such that for any $g \in G$, $g = th$ for some $t \in T$ and $h \in H$.

We define an operator Test so that $\mathsf{Test} = \mathsf{Test_r} \cdot \cdots \cdot \mathsf{Test_2} \cdot \mathsf{Test_1}$, where each unitary operator $\mathsf{Test_i}$ tests whether f is H_i-periodic. Each $\mathsf{Test_i}$ is defined by $\mathsf{Test_i} = Q_i \otimes P_{s,i} + I \otimes P_{s,i}^{\perp}$ where

$$Q_i : \quad \left\{ \begin{array}{lcll} |0\rangle\,|0\rangle & \mapsto & |i\rangle\,|1\rangle & \\ |v\rangle\,|l\rangle & \mapsto & |v\rangle\,|l+1\rangle\,, & \quad \text{if } l > 0 \end{array} \right. \qquad \text{and } P_{s,i} = \left(\sum_{t \in T_i} |tH_i\rangle \langle tH_i| \otimes I \right)^{\otimes s}$$

Here Q_i acts on the first two registers so that once the second register is increased from 0 to 1, the first register stays the same, and $P_{s,i}$ is the projector of the s couplets. The effect of $\mathsf{Test_i}$ is that Q_i is applied on the first two registers if s subgroup registers are in coset states of H_i.

We create an initial state to be

$$|\Psi_{init}\rangle = |0\rangle\,|0\rangle \otimes \left(\frac{1}{\sqrt{N}} \sum_{g \in G} |g\rangle\,|f(g)\rangle \right)^{\otimes s}$$

by s many query calls.

[3] if a function f is H-periodic then it is also H'-periodic for a proper subgroup H' of H. So we want to test for bigger subgroups first.

[4] We know that the number of r is $2^{O(n^2)}$ since any H_i is generated by a set of at most n elements of G

Lemma 1. *If f is H_i-periodic, then*

$$\mathsf{Test_i} \left|\Psi_{init}\right\rangle = \left|i\right\rangle \left|1\right\rangle \otimes \left(\frac{1}{\sqrt{N}} \sum_{g\in G} \left|g\right\rangle \left|f(g)\right\rangle \right)^{\otimes s}$$

Proof. First, we realize that if f is H_i-periodic then s subgroup registers are in superposition of coset states $\left|tH_i\right\rangle = \frac{1}{\sqrt{|H_i|}} \sum_{h\in H_i} \left|th\right\rangle$ for $t \in T_i$, a translation of H_i. Also, f being H_i-periodic implies that $f(t) = f(th)$ for all $t \in T_i$ and $h \in H_i$. So the state $\frac{1}{\sqrt{N}} \sum_{g\in G} \left|g\right\rangle \left|f(g)\right\rangle = \frac{1}{\sqrt{N}} \sum_{t\in T_i} \left|tH_i\right\rangle \left|f(g)\right\rangle$ lives in +1-eigenspace of $P_{1,i}$, and hence $P_{s,i}$ leaves the s couplets untouched. Thus, the lemma follows. □

At the end of the day, what we want to do is to apply the unitary Test to the initial state $\left|\Psi_{init}\right\rangle$ to get $\left|\Psi_{final}\right\rangle$. Then, we measure the first register of $\left|\Psi_{final}\right\rangle$ to get the subgroup index v, and if $1 \leq v \leq r$ we output a generating set for H_i, otherwise we output 1_G. But our output may be a wrong answer.
As we iterate through r tests for the subgroups, we would wish to only alter the state marginally so that it is safe to continue to test for the next H_{i+1} subgroup.

Lemma 2. *If f is not H_i-periodic, then the distance $\left|(\mathsf{Test_i} \left|\Psi_{init}\right\rangle) - \left|\Psi_{init}\right\rangle \right|$ is at most $\frac{2}{2^{s/2}}$.*

The next lemma follows since distances add up linearly.

Lemma 3. *If f is not H_i-periodic for any $1 \leq i \leq j$, then the distance $\left| \left|\Psi_j\right\rangle - \left|\Psi_{init}\right\rangle \right|$ is at most $\frac{2j}{2^{s+2}}$.*

At the beggining in the Theorem, we stated that the error probability is exponential. Great news is that we can make the algorithm exact by the use of classical precomputing and amplitude amlification.
First, we partition the set of subgroups $\{H_1, H_2, \ldots, H_r\}$ into $Y_{1/4}$ and $Y_{3/4}$. We'll look at a new algorithm called $\mathsf{ExactTest}$ that identifies which of the partition the hidden subgroup belongs to.

Lemma 4. *Consider measuring the ancilla qubit of the state $\mathsf{ExactTest}(\left|\Psi_{init}\right\rangle \otimes \left|0\right\rangle)$. The probability that the outcome of this measurement is 1 is 3/4 if the hidden subgroup H is in $Y_{3/4}$, and it is 1/4 if H is in $Y_{1/4}$.*

In order to describe what $\mathsf{ExactTest}$ is, we need to define few things. Let $Prob[H_i|H]$ be the probability that the measurement outcome of the first register of $\left|\Psi_{final}\right\rangle$ is i, conditioned on the hidden subgroup being H. Let M be an $r \times r$ matrix over $[0, 1]$ with each row and column indexed by a subgroup. Let entry (H, H_i) of M be $Prob[H_i|H]$. Let y be an $r \times 1$ vector with entries from $\{1/4, 3/4\}$ and each entry indexed by a subgroup. Let $x = M^{-1}y$. Every entry of x is in $[0, 1]$ for $r \geq 4$.
Now we define $\mathsf{ExactTest}$

$$\mathsf{ExactTest} = \mathsf{R} \cdot (\mathsf{Test} \otimes \mathsf{I})$$

When we apply $\mathsf{ExactTest}$ to the state $|\Psi_{init}\rangle \otimes |0\rangle$, Test is applied to the state $|\Psi_{init}\rangle$ and leaves the ancilla unchanged. Then, R is applied, which is unitary that transforms the ancilla qubit $|0\rangle$ into $\sqrt{1-x_i}\,|0\rangle + \sqrt{x_i}\,|1\rangle$, conditioned on the output register holding the subgroup index i. Thus, the probability that the measurement outcome of the ancilla qubit being 1 is $\sum_i x_i Prob[H_i|H_v]$, which is precisely the entry y_v of y, where H_v is the hidden subgroup. So we just need to set $y_v = 3/4$ if $H_v \in Y_{3/4}$ or set $y_v = 1/4$ if $H_v \in Y_{1/4}$. This proves the lemma
The lemma showed us that we can distinguish between the two sets $Y_{3/4}$ and $Y_{1/4}$ with probabilites 3/4 and 1/4, respectively. Now by using amplitude amplification, we can alter these probabilities into being 0 and 1. Hence, we can distiguish between the two sets. Applying binary search on the set of subgroups, varying the choices of $Y_{3/4}$ and $Y_{1/4}$, we can find the hidden subgroup with certainty.

3 Shor's algorithm reduct to HSP problem of $\mathbb{Z}_{\mathbb{N}}$

Shor's algorithm reduce the problem of factorization of a composite natural number $\mathbb{N}$ to finding the order of an arbitrary non-identity element in $\mathbb{Z}_{\mathbb{N}}$.

Definition 4. A group $\mathbb{Z}_{\mathbb{N}}$ is the set of remainders of a natural number N, up to congruence class.

Definition 5. Order of an element a, where $gcd(a, N) = 1$, in $\mathbb{Z}_{\mathbb{N}}$ is defined by the smallest natural number r, that $a^r \equiv 1mod_N$.

Then by Lagrange's theorem, the order of all units in $\mathbb{Z}_{\mathbb{N}}$ divide $\phi(N)$, the Euler function value of N, that is less than N, when N is greater than one. The Shor's algorithm begin with choose an arbitrary number in $a \in \{2, ...,$N-1$\}$, then compute the $gcd(a, N)$, if it is not 1, then we already have a non-trivial divisor of N, if it is not one, then we apply it to oder finding algorithm to find the order.Then we get:

$$\begin{aligned} a^r &\equiv 1mod_N \\ &\Rightarrow (a^{\frac{r}{2}-1})(a^{\frac{r}{2}}+1) \end{aligned}$$

Then if $N|a^{\frac{r}{2}-1}$, implies that $a^{\frac{r}{2}-1}mod_N \equiv 1mod_N$, then r is not the period, this cannot happen. Then N must divide $a^{\frac{r}{2}+1}$, then compute $gcd(a^{\frac{r}{2}-1}, N)$ to get a non-trivial divisor of N.

In this process the quantum order finding algorithm finds the period of this function:

$$f : \mathbb{N} \to \mathbb{N}$$

Then we have a function $f(a) = x^a\ mod_N$ and $f(a) = f(b)$ iff $a = b + rk$, where k is an arbitrary integer. Thus it hides the subgroup of of $\mathbb{Z}_{\mathbb{N}}$ which is generated by r. If the function $f : \mathbb{N} \to \mathbb{N}$ has period r, then $f(a) = f(b)$ iff a and b are in the same coset generated by r, i.e. $a \in b+ < r >$.

Moreover, this type of hidden subgroup can also be applied to compute discrete logarithms.

Definition 6. An unit in ring $\mathbb{Z}_p$ is any element a$\in \mathbb{Z}_p$, such that this is an element b where $a * b = 1$

Consider a prime p, in the group $\mathbb{Z}_p$, the units form a cyclic group of oder $p-1$, that is $\mathbb{Z}_p^\times$. Then, suppose there is two element a and b, such that $a = b^r mod_p$, for same r, and discrete logarithm is to compute r. That is to translate to the HSP in abelian group $\mathbb{Z}_p^\times \times \mathbb{Z}_p^\times$, where the kernel of the function is the generating set of the hidden subgroup and also the discrete logarithm power.

$$f(a, b) = g^a x^{-b} mod_p$$

And the hidden subgroup is generated by $(r, 1)$

4 Dihedral Group HSP

4.1 Kuperberg's Algorithm

Definition 7. The *Dihedral Group* of order $2N$

$$D_N = \{r, s | ord(r) = N, ord(s) = 2, srs = r^{-1}\}.$$

This group can be thought of as the symmetries of a regular N-gon, with the rotation r and the flip s.

Now, to solve the HSP on the Dihedral Group, we first need to characterize the subgroups of D_N. There are three such classes of groups:

- The cyclic subgroups generated by some k, where $k \in \mathbb{Z}_N$. These subgroups are normal in D_N.
- The order 2 ”flip” subgroups generated by some sr^k, with some k defined as above.
- The dihedral subgroups D_m, where $m|N$. Each of these is a semidirect product of one subgroup of each previous type.

In 1998, Ettinger and Hoyer showed that it is possible to reduce the problem of solving the hidden subgroup problem on D_N with hiding function f, to calculating k assuming the hidden subgroup is generated by sr^k.

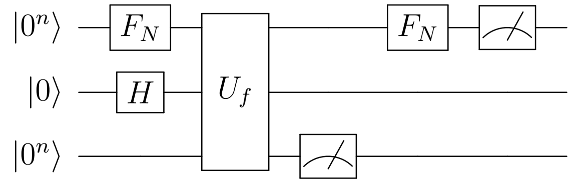
Theorem 8. *Solving the Hidden Subgroup problem on D_N can be reduced to detecting a hidden order 2 subgroup sr^k, or detecting that no such subgroup is hidden.*

Proof. If the hidden subgroup, call it H, is not one of the ”flip” subgroups, then it must have a cyclic part. That is, it itself must have a cyclic subgroup, $H \cap Z_N$. This subgroup can be efficiently detected - it is normal in the dihedral group. Suppose we then determine that $H \cap Z_N \simeq Z_k$. We then do the following: Compute the factor group $D_N/Z_k \simeq D_{N/k}$. Then the subgroup of $D_{N/k}$ which will be hidden by the function is just H/Z_k, which has no cyclic subgroup. As such, it can only be a flip or the trivial subgroup. Then we have reduced the problem to detecting either the specific flip that was done or the trivial subgroup on this smaller group. □

The standard procedure for solving HSP, used in Shor's algorithm for $\mathbb{Z}_N$, Simon's algorithm mod p (from our homework) over $\mathbb{Z}_p \times \mathbb{Z}_p$, and others, is to obtain a superposition over all possible states, then apply function unitary and apply the inverse of the unitary that gave us the superposition of all states. This procedure is no different for Kuperberg's method for solving HSP on the dihedral group. We represent the element

$$s^b r^a \in D_N : |a\rangle\,|b\rangle\,,\text{with } b \in \{0,1\}, a \in \mathbb{Z}_N.$$

Then a will take $n = \lceil \log(N) \rceil$ qubits to represent, and b will take 1. We also want to be able to store output, so we represent output in another n qubits, so we start with $|0^n\rangle\,|0\rangle\,|0^n\rangle$. The circuit we apply is



where our unitary U_f takes $|a\rangle\,|b\rangle\,|c\rangle \to |a\rangle\,|b\rangle\,|c + f(a,b)\rangle$. We also have F_N is the quantum Fourier transform for $\mathbb{Z}_N$. After applying the quantum circuit, we measure the function output.

Theorem 9. *After the first measurement, we have a quantum state of the form* $\frac{1}{\sqrt{2}}(|x,0\rangle + |(x+d) mod\ M, 1\rangle)$*, where the function hides the subgroup* H*, generated by a flip* sr^d*.*

Proof. After the Quantum Fourier Transform and Hadamard, we are in state

$$\frac{1}{\sqrt{2N}} \sum_{g \in D_N} |g\rangle\,|0\rangle\,,$$

i.e. a uniform superposition of all group members. We then apply the unitary to get

$$\frac{1}{\sqrt{2N}} \sum_{g \in D_N} |g\rangle\,|f(g)\rangle\,.$$

Now, measuring the last state will collapse to only states with the measured $f(g)$. That is, the group qubits will be in states g with fixed $f(g)$. They will form a coset of H. Because we assumed that H consists of $\{s^0 r^0, sr^d\}$, the group qubits will be in the state $\frac{1}{\sqrt{2}}(|x,0\rangle + |(x+d) \text{mod } M, 1\rangle)$. □

Theorem 10. *At the end of the quantum circuit, we have some state* $\frac{1}{\sqrt{2}}(|0\rangle + \omega^{dy}|1\rangle)$*, where* $\omega = e^{2\pi i/N}$*, and* $y \in Z_N$*.*

Proof. From the state $\frac{1}{\sqrt{2}}(|x, 0\rangle + |(x+d)\text{mod } M, 1\rangle)$, applying a Quantum Fourier Transform will provide an even distribution over each value in Z_N, given by:

$$\frac{1}{\sqrt{2N}} \sum_{y \in Z_N} (\omega^{xy}|y, 0\rangle + \omega^{(x+d)y}|y, 1\rangle).$$

Then we can measure the n qubit register, which will collapse it to some value y, and the remaining flip qubit will be in the state $\frac{1}{\sqrt{2}}\omega^{xy}(|0\rangle + \omega^{dy}|1\rangle)$. □

If it was easy to construct a matrix with this as an eigenvector, we could just run phase estimation. Unfortunately, it appears that this would require knowledge of d, which is the exact quantity to be identified. Unfortunately, we don't have this, so we tensor two of these qubits:

$$\frac{1}{2}(|0\rangle + \omega^{dy_1}|1\rangle)(|0\rangle + \omega^{dy_2}|1\rangle),$$

and measure the parity of this qubit, hoping to get the odd case:

$$\frac{1}{2}(\omega^{dy_1}|10\rangle + \omega^{dy_2}|01\rangle) \to \frac{1}{2}\omega^{dy_1}(|10\rangle + \omega^{dy_2 - dy_1}|01\rangle).$$

This can be treated as a qubit just of the form

$$\frac{1}{\sqrt{2}}(|0\rangle + \omega^{dy_2 - dy_1}|1\rangle),$$

so there is a way constructing combinations of these samples with phases on the 1 state that are not uniformly random. This construction can be used to, in $2^O(\sqrt{\log N})$ time, create the state

$$\frac{1}{\sqrt{2}}(|0\rangle + \omega^{d2^N}|1\rangle),$$

which is just $|+\rangle$ if d is even, and $|-\rangle$ if d is odd. We measure in the Hadamard basis and receive the least significant bit of d. Then this process can be iterated on smaller groups $D_{N/2}$ until d is expressed completely. This will take about $\log(N)$ iterations.

4.2 Shortest Vector Problem

To discuss how the Dihedral HSP can be used to solve the shortest vector problem, we introduce some terms.

Definition 11. A *lattice* is a discrete subgroup of $\mathbb{R}^n$. It can also be considered to be all integer linear combinations of a set of basis vectors.

Definition 12. The *shortest vector problem* is the problem to, given a lattice, find the shortest nonzero vector in the lattice. In this section, we will call the length of this vector λ_1, and the shortest vector $\mathbf{w}$.

However, the actual problem that will be solved by the quantum algorithm is not the SVP. It is a variant:

Definition 13. The $f(n)-$*unique SVP* is the problem to, given a lattice, find the shortest nonzero vector in it. The lattice is guaranteed to have that only multiples of the shortest vector have length shorter than the $f(n)\lambda_1$.

This weaker problem is not arbitrary - it is the type of condition that is commonly suggested for quantum-resistant post-quantum cryptographic protocols.

The way that the Dihedral HSP is used to solve this problem uses what is called the *two-point problem.* Consider $\mathbb{R}^n$ partitioned into cubes, such that there are only two lattice vectors in each cube. The cubes should be small enough that the only two vectors are separated by the shortest vector $\mathbf{w}$. This is where the weakened problem is relevant - it allows for a wider range of cube widths. We also may consider a subset of the lattice vectors, to ensure we have only 2 lattice vectors in the cube. Now, the quantum algorithm is to produce this lattice subset partitioned into cubes, then to measure and collapse the state to one such cube. It then can find the shortest vector. For the precise specifications, see [3].

Theorem 14. *Suppose we have a procedure to produce superpositions of vectors of the form* $\frac{1}{\sqrt{2}}(|0,\mathbf{a}\rangle + |1,\mathbf{a}'\rangle)$*, where* $\mathbf{a}, \mathbf{a}'$ *are lattice vectors separated by the shortest vector* $\mathbf{w}$*. The lattice vectors are of the form* $\{0, 1, \cdots M-1\}^n$ *for some* M, n*. We can produce the shortest vector by reducing the two point problem to the Dihedral HSP using this procedure.*

Proof. The general procedure will be to convert the lattice vectors into integers that can be examined as an instance of the Dihedral HSP. We will do this with a function $f : \mathbb{Z}_M^n \to \mathbb{Z}_{(2M)^n}$:

$$f([a_1, a_2, \cdots, a_n]^T) = \sum_{i=1}^{n} (2M)^{i-1} a_i.$$

This function is one-to-one, so consider it as a unitary, which we apply to the superpositions $\frac{1}{\sqrt{2}}(|0,\mathbf{a}\rangle + |1,\mathbf{a}'\rangle)$ to get

$$\frac{1}{\sqrt{2}}(|0,\mathbf{a}\rangle + |1,\mathbf{a}'\rangle) \to \frac{1}{\sqrt{2}}(|0,\mathbf{f}(\mathbf{a})\rangle + |1,\mathbf{f}(\mathbf{a}')\rangle).$$

If we apply this to each of the quantum registers, the difference $f(\mathbf{a}') - f(\mathbf{a})$ is a fixed value, because it represents the shortest vector in each case, and the function is one-to-one. Then this is a well defined value to give as an input to Kuperberg's Dihedral HSP algorithm, on $D_{(2M)^n}$. Then we can achieve the output $f(\mathbf{a}') - f(\mathbf{a})$ using a sufficient amount of these

samples.
Then our goal now is to take $f(\mathbf{a}') - f(\mathbf{a})$ and output $\mathbf{a}' - \mathbf{a}$. Our output is of the form

$$\sum_{i=1}^{n} (2M)^{i-1} b_i : b_i = a_i' - a_i \in \{-M, -M+1, \cdots M-1\}.$$

In order to extract the b_i values we compute

$$\sum_{i=1}^{n} (2M)^{i-1} b_i + \sum_{i=1}^{n} M(2M)^{i-1} = \sum_{i=1}^{n} (2M)^{i-1}(b_i + M),$$

from which it is straightforward to compute the $b_i + M \in \mathbb{Z}_{\not\mathbb{2}\mathbb{M}}$, and from there we can compute each $b_i = a_i' - a_i$. The vector $\mathbf{b} = [b_1, b_2, \cdots, b_n]^T$ is the shortest vector, the solution to the two point problem. □

Recall that the Dihedral HSP on D_N can be solved in $2^{O(\sqrt{\log(N)})}$ time using Kuperberg's algorithm. Then on $D_{(2M)^n}$, the time is $2^{O(\sqrt{\log((2M)^n)})} = 2^{O(\sqrt{n \log(2M)})}$, which seems to be roughly $2^{O(\sqrt{x})}$ with x as the size of the input. However, there is a quadratic increase in the Regev algorithm for SVP that causes this to actually be a $2^{O(n)}$ algorithm, equivalent in runtime to the best known classical algorithms.

References

[1] Kirsten Eisenträger, Sean Hallgren, Alexei Kitaev, and Fang Song. *A quantum algorithm for computing the unit group of an arbitrary degree number field.* 2014 ACM Symposium on Theory of Computing, 2014.

[2] Oded Regev. *A Subexponential Time Algorithm for the Dihedral Hidden Subgroup Problem with Polynomial Space.* arXiv:quant-ph/0406151, 2004.

[3] Oded Regev. *Quantum Computation and Lattice Problems.* arXiv:cs/0304005, 2003.

[4] Mark Ettinger, Peter Hoyer, and Emanuel Knill. *The quantum query complexity of the hidden subgroup problem is polynomial.* arXiv:quant-ph/0401083, 2004.

[5] Mark Ettinger, Peter Hoyer. *On Quantum Algorithms for Noncommutative Hidden Subgroups.* arXiv:quant-ph/9807029, 1998.

[6] Greg Kuperberg. *A subexponential-time quantum algorithm for the dihedral hidden subgroup problem.* SIAM Journal on Computing, 2005.

[7] Frédéric Wang. *The Hidden Subgroup Problem.* arXiv:1008.0010, 2010.